

# **X Window System C Quick Reference Guide**

**HP 9000 Series 300/800 Computers**

HP Part Number 98594-90070



**HEWLETT  
PACKARD**

**Hewlett-Packard Company**

3404 East Harmony Road, Fort Collins, Colorado 80525

---

## Legal Notices

The information contained in this document is subject to change without notice.

*Hewlett-Packard makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.* Hewlett-Packard shall not be liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

**Warranty.** A copy of the specific warranty terms applicable to your Hewlett-Packard product and replacement parts can be obtained from your local Sales and Service Office.

Copyright © Hewlett-Packard Company 1988

This document contains information which is protected by copyright. All rights are reserved. Reproduction, adaptation, or translation without prior written permission is prohibited, except as allowed under the copyright laws.

**Restricted Rights Legend.** Use, duplication or disclosure by the U.S. Government Department of Defense is subject to restrictions as set forth in paragraph (b)(3)(ii) of the Rights in Technical Data and Software clause in FAR 52.227-7013.

Use of this manual and flexible disc(s) or tape cartridge(s) supplied for this pack is restricted to this product only. Additional copies of the programs can be made for security and back-up purposes only. Resale of the programs in their present form or with alterations, is expressly prohibited.

Copyright © AT&T, Inc. 1980, 1984

Copyright © The Regents of the University of California 1979, 1980, 1983

This software and documentation is based in part on the Fourth Berkeley Software Distribution under license from the Regents of the University of California.

---

## Printing History

The manual printing date and part number indicate its current edition. The printing date will change when a new edition is printed. Minor changes may be made at reprint without changing the printing date. The manual part number will change when extensive changes are made.

Manual updates may be issued between editions to correct errors or document product changes. To ensure that you receive these updates or new editions, you should subscribe to the appropriate product support service. See your HP sales representative for details.

December 1988 ... Edition 1.

---

## Introduction

The *X Window System C Quick Reference Guide* provides a list of the C functions and widgets that are available to the users of the X Window System through Xlib, the HP X Widgets, and the Xt Intrinsics.

Xlib is a C subroutine library that application programs (clients) use to interface with the window system. The HP X Widgets provide a base of predefined features that the programmer can use to build an application. A widget programmer can design new widgets by combining existing widgets or by using the Xt Intrinsics.

For detailed information about each function or widget, refer to *Programming With Xlib*, *Programming With the HP X Widgets*, and *Programming With the Xt Intrinsics*.

This quick reference guide is organized as follows:

- |           |  |
|-----------|--|
| Chapter 1 | Functions alphabetized by function name. Each entry contains the function call, parameter types, and the name of the manual containing more information.             |
| Chapter 2 | Functions alphabetized by description. Each entry contains a description of the function, the function call, and the name of the manual containing more information. |
| Chapter 3 | Widgets alphabetized by widget name. Each entry contains a description of the widget, its Widget class, its class name, and the resources available to it.           |



## Functions Sorted by Function Name

---

The items in this table are in alphabetical order, disregarding the prefixes, such as Xt, Xrm, Xw, and the type, such as void or int.

Function	Manual
XHPAcknowledge( <i>display</i> , <i>deviceid</i> , <i>acknowledge</i> ) Display * <i>display</i> ; XID <i>deviceid</i> ; unsigned int <i>acknowledge</i> ;	Intrinsics
void ActionProc( <i>w</i> , <i>event</i> , <i>params</i> , <i>num_params</i> )	Intrinsics
XActivateScreenSaver( <i>display</i> ) Display * <i>display</i> ;	Xlib
void XtAddActions( <i>actions</i> , <i>num_actions</i> ) XtActionList <i>actions</i> ; Cardinal <i>num_actions</i> ;	Intrinsics
void XtAddCallback( <i>w</i> , <i>callback_name</i> , <i>callback</i> , <i>client_data</i> ) Widget <i>w</i> ; String <i>callback_name</i> ; XtCallbackProc <i>callback</i> ; caddr_t <i>client_data</i> ;	Intrinsics
void XtAddCallbacks( <i>w</i> , <i>callback_name</i> , <i>callbacks</i> ) Widget <i>w</i> ; String <i>callback_name</i> ; XtCallbackList <i>callbacks</i> ;	Intrinsics
void XtAddConverter( <i>from_type</i> , <i>to_type</i> , <i>converter</i> , <i>convert_args</i> , <i>num_args</i> ) String <i>from_type</i> , <i>to_type</i> ; XtConverter <i>converter</i> ; XtConvertArgList <i>convert_args</i> ; Cardinal <i>num_args</i> ;	Intrinsics

Function	Manual
<b>void XtAddEventHandler(<i>w, event_mask, nonmaskable, proc, client_data</i>)</b> Widget <i>w</i> ; EventMask <i>event_mask</i> ; Boolean <i>nonmaskable</i> ; XtEventHandler <i>proc</i> ; caddr_t <i>client_data</i> ;	Intrinsics
<b>void XtAddExposureToRegion(<i>event, region</i>)</b> XEvent * <i>event</i> ; Region <i>region</i> ;	Intrinsics
<b>void XtAddGrab(<i>w, exclusive, spring_loaded</i>)</b> Widget <i>w</i> ; Boolean <i>exclusive, spring_loaded</i> ;	Intrinsics
<b>XAddHost(<i>display</i>)</b> Display * <i>display</i> ;	Xlib
<b>XAddHosts(<i>display, hosts, num_hosts</i>)</b> Display * <i>display</i> ; XHostAddress * <i>hosts</i> ; int <i>num_hosts</i> ;	Xlib
<b>XtInputId XtAddInput(<i>source, condition, proc, client_data</i>)</b> int <i>source</i> ; caddr_t <i>condition, client_data</i> ; XtInputCallbackProc <i>proc</i> ;	Intrinsics
<b>XAddPixel(<i>ximage, value</i>)</b> XImage * <i>ximage</i> ; long <i>value</i> ;	Xlib
<b>void XtAddRawEventHandler(<i>w, event_mask, nonmaskable, proc, client_data</i>)</b> Widget <i>w</i> ; EventMask <i>event_mask</i> ; Boolean <i>nonmaskable</i> ; XtEventHandler <i>proc</i> ; caddr_t <i>client_data</i> ;	Intrinsics
<b>XtIntervalId XtAddTimeOut(<i>interval, proc, client_data</i>)</b> unsigned long <i>interval</i> ; XtTimerCallbackProc <i>proc</i> ; caddr_t <i>client_data</i> ;	Intrinsics

Function	Manual
<b>XAddToExtensionList</b> ( <i>structure, ext_data</i> ) struct _XExtData ** <i>structure</i> ; XExtData * <i>ext_data</i> ; 	Xlib
<b>XAddToSaveSet</b> ( <i>display, w</i> ) Display * <i>display</i> ; Window <i>w</i> ; 	Xlib
<b>XtWorkProcId XtAddWorkProc</b> ( <i>proc, closure</i> ) XtWorkProc <i>proc</i> ; Opaque <i>closure</i> ; 	Intrinsics
<b>Status XAllocColor</b> ( <i>display, cmap, screen_in_out</i> ) Display * <i>display</i> ; Colormap <i>cmap</i> ; XColor * <i>screen_in_out</i> ; 	Xlib
<b>Status XAllocColorCells</b> ( <i>display, cmap, contig, plane_masks_return,</i> <i>nplanes, pixels_return, ncolors</i> ) Display * <i>display</i> ; Colormap <i>cmap</i> ; Bool <i>contig</i> ; unsigned long <i>plane_masks_return</i> [ ], <i>pixels_return</i> [ ]; unsigned int <i>nplanes, ncolors</i> ; 	Xlib
<b>Status XAllocColorPlanes</b> ( <i>display, cmap, contig, pixels_return, ncolors,</i> <i>nreds, ngreens, nblues, rmask_return, gmask_return, bmask_return</i> ) Display * <i>display</i> ; Colormap <i>cmap</i> ; Bool <i>contig</i> ; unsigned long <i>pixels_return</i> [ ]; int <i>ncolors, nreds, ngreens, nblues</i> ; unsigned long * <i>rmask_return, *gmask_return, *bmask_return</i> ; 	Xlib
<b>Status XAllocNamedColor</b> ( <i>display, cmap, color_name, screen_def_return,</i> <i>exact_def_return</i> ) Display * <i>display</i> ; Colormap <i>cmap</i> ; char * <i>color_name</i> ; XColor * <i>screen_def_return, *exact_def_return</i> ; 	Xlib



Function	Manual
XAllowEvents( <i>display, event_mode, time</i> ) Display * <i>display</i> ; int <i>event_mode</i> ; Time <i>time</i> ;	Xlib
unsigned long XAllPlanes( )	Xlib
XtInputId XtAppAddInput( <i>app_context, source, condition, proc, client_data</i> ) XtAppContext <i>app_context</i> ; int <i>source</i> ; caddr_t <i>condition, client_data</i> ; XtInputCallbackProc <i>proc</i> ;	Intrinsics
XtIntervalId XtAppAddTimeOut( <i>app_context, interval, proc, client_data</i> ) XtAppContext <i>app_context</i> ; unsigned long <i>interval</i> ; XtTimerCallbackProc <i>proc</i> ; caddr_t <i>client_data</i> ;	Intrinsics
XtWorkProcId XtAppAddWorkProc( <i>app_context, proc, client_data</i> ) XtAppContext <i>app_context</i> ; XtWorkProc <i>proc</i> ; caddr_t <i>client_data</i> ;	Intrinsics
Widget XtAppCreateShell( <i>application_name, application_class, widget_class, display, args, num_args</i> ) String <i>application_name, application_class</i> ; WidgetClass <i>widget_class</i> ; Display * <i>display</i> ; Arglist <i>args</i> ; Cardinal <i>num_args</i> ;	Intrinsics
void XtAppMainLoop( <i>app_context</i> ) XtAppContext <i>app_context</i> ;	Intrinsics
void XtAppNextEvent( <i>app_context, event_return</i> ) XtAppContext <i>app_context</i> ; XEvent * <i>event_return</i> ;	Intrinsics
Boolean XtAppPeekEvent( <i>app_context, event_return</i> ) XtAppContext <i>app_context</i> ; XEvent * <i>event_return</i> ;	Intrinsics



Function	Manual
Boolean XtAppPending( <i>app_context</i> ) XtAppContext <i>app_context</i> ;	Intrinsics
void XtAppProcessEvent( <i>app_context</i> , <i>mask</i> ) XtAppContext <i>app_context</i> ; XtInputMask <i>mask</i> ;	Intrinsics
XwarrowWidgetClass	Widgets
void XtAugmentTranslations( <i>w</i> , <i>translations</i> ) Widget <i>w</i> ; XtTranslations <i>translations</i> ;	Intrinsics
XAutoRepeatOff( <i>display</i> ) Display * <i>display</i> ;	Xlib
XAutoRepeatOn( <i>display</i> ) Display * <i>display</i> ;	Xlib
XBell( <i>display</i> , <i>percent</i> ) Display * <i>display</i> ; int <i>percent</i> ;	Xlib
int XBitmapBitOrder( <i>display</i> ) Display * <i>display</i> ;	Xlib
int XBitmapPad( <i>display</i> ) Display * <i>display</i> ;	Xlib
int XBitmapUnit( <i>display</i> ); Display * <i>display</i> ;	Xlib
unsigned long XBlackPixel( <i>display</i> , <i>screen_number</i> ) Display * <i>display</i> ; int <i>screen_number</i> ;	Xlib
unsigned long XBlackPixelOfScreen( <i>screen</i> ) Screen * <i>screen</i> ;	Xlib
void XtBuildEventMask( <i>w</i> ) Widget <i>w</i> ;	Intrinsics
XwbulletinWidgetClass	Widgets
XwbuttonWidgetClass	Widgets

Function	Manual
void XtCallbackExclusive( <i>w, client_data, call_data</i> ) Widget <i>w</i> ; caddr_t <i>client_data, call_data</i> ;	Intrinsics
void XtCallbackNone( <i>w, client_data, call_data</i> ) Widget <i>w</i> ; caddr_t <i>client_data, call_data</i> ;	Intrinsics
void XtCallbackNonexclusive( <i>w, client_data, call_data</i> ) Widget <i>w</i> ; caddr_t <i>client_data, call_data</i> ;	Intrinsics
void XtCallbackPopdown( <i>w, client_data, call_data</i> ) Widget <i>w</i> ; caddr_t <i>client_data, call_data</i> ;	Intrinsics
void XtCallCallbacks( <i>w, callback_name, call_data</i> ) Widget <i>w</i> ; String <i>callback_name</i> ; caddr_t <i>call_data</i> ;	Intrinsics
char *XtCalloc( <i>num, size</i> ) Cardinal <i>num, size</i> ;	Intrinsics
XwcascadeWidgetClass	Widgets
int XCellsOfScreen( <i>screen</i> ) Screen * <i>screen</i> ;	Xlib
XChangeActivePointerGrab( <i>display, event_mask, cursor, time</i> ) Display * <i>display</i> ; unsigned int <i>event_mask</i> ; Cursor <i>cursor</i> ; Time <i>time</i> ;	Xlib
XHPChangeDeviceControl( <i>display, deviceid, value_mask, values</i> ) Display * <i>display</i> ; XID <i>deviceid</i> ; unsigned long <i>value_mask</i> ; XHPDeviceControl * <i>values</i> ;	Xlib

Function	Manual
<hr/> XHPChangeDeviceKeyMapping( <i>display, deviceid, first_keycode, keysyms_per_keycode, keysyms, num_codes</i> ) Display *display; XID deviceid; int first_keycode, keysyms_per_keycode, num_codes; KeySyms *keysyms; <hr/>	Xlib
XChangeGC( <i>display, gc, valuemask_change, values</i> ) Display *display; GC gc; unsigned long valuemask_change; XGCValues *values; <hr/>	Xlib
XChangeKeyboardControl( <i>display, value_mask, values</i> ) Display *display; unsigned long value_mask; XKeyboardControl *values; <hr/>	Xlib
XChangeKeyboardMapping( <i>display, first_keycode, keysyms_per_keycode, keysyms, num_codes</i> ) Display *display; int first_keycode, keysyms_per_keycode; KeySym *keysyms; int num_codes; <hr/>	Xlib
XChangePointerControl( <i>display, do_accel, do_threshold, accel_numerator, accel_denominator, threshold</i> ) Display *display; Bool do_accel, do_threshold; int accel_numerator, accel_denominator; int threshold; <hr/>	Xlib
XChangeProperty( <i>display, w, property, type, format, mode, data, nelements</i> ) Display *display; Window w; Atom property, type; int format, mode, nelements; unsigned char *data; <hr/>	Xlib



Function	Manual
<hr/> XChangeSaveSet( <i>display, w, change_mode</i> ) Display * <i>display</i> ; Window <i>w</i> ; int <i>change_mode</i> ; <hr/>	Xlib
XChangeWindowAttributes( <i>display, w, valuemask, attributes</i> ) Display * <i>display</i> ; Window <i>w</i> ; unsigned long <i>valuemask</i> ; XSetWindowAttributes * <i>attributes</i> ; <hr/>	Xlib
Bool XCheckIfEvent( <i>display, event_return, predicate, arg</i> ) Display * <i>display</i> ; XEvent * <i>event_return</i> ; Bool (* <i>predicate</i> )(); char * <i>arg</i> ; <hr/>	Xlib
Bool XCheckMaskEvent( <i>display, event_mask, event_return</i> ) Display * <i>display</i> ; long <i>event_mask</i> ; XEvent * <i>event_return</i> ; <hr/>	Xlib
void XtCheckSubclass( <i>w, widget_class, message</i> ) <hr/>	Intrinsics
Bool XCheckTypedEvent( <i>display, event_type, event_return</i> ) Display * <i>display</i> ; int <i>event_type</i> ; XEvent * <i>event_return</i> ; <hr/>	Xlib
Bool XCheckTypedWindowEvent( <i>display, w, event_type, event_return</i> ) Display * <i>display</i> ; Window <i>w</i> ; int <i>event_type</i> ; XEvent * <i>event_return</i> ; <hr/>	Xlib
XCheckWindowEvent( <i>display, w, event_mask, event_return</i> ) Display * <i>display</i> ; Window <i>w</i> ; long <i>event_mask</i> ; XEvent * <i>event_return</i> ; <hr/>	Xlib



Function	Manual
XCirculateSubwindows( <i>display, w, direction</i> ) Display * <i>display</i> ; Window <i>w</i> ; int <i>direction</i> ;	Xlib
XCirculateSubwindowsDown( <i>display, w</i> ) Display * <i>display</i> ; Window <i>w</i> ;	Xlib
XCirculateSubwindowsUp( <i>display, w</i> ) Display * <i>display</i> ; Window <i>w</i> ;	Xlib
WidgetClass XtClass( <i>w</i> ) Widget <i>w</i> ;	Intrinsics
XClearArea( <i>display, w, x, y, width, height, exposures</i> ) Display * <i>display</i> ; Window <i>w</i> ; int <i>x, y</i> ; unsigned int <i>width, height</i> ; Bool <i>exposures</i> ;	Xlib
XClearWindow( <i>display, w</i> ) Display * <i>display</i> ; Window <i>w</i> ;	Xlib
XClipBox( <i>r, rect_return</i> ) Region <i>r</i> ; XRectangle * <i>rect_return</i> ;	Xlib
XCloseDisplay	Intrinsics
XCloseDisplay( <i>display</i> ) Display * <i>display</i> ;	Xlib
void XtCloseDisplay( <i>display</i> ) Display <i>display</i> ;	Intrinsics

Function	Manual
<b>void XtConfigureWidget(<i>w, x, y, width, height, border_width</i>)</b> Widget <i>w</i> ; Position <i>x, y</i> ; Dimension <i>width, height, border_width</i> ;	Intrinsics
<b>XConfigureWindow(<i>display, w, value_mask, values</i>)</b> Display <i>*display</i> ; Window <i>w</i> ; unsigned int <i>value_mask</i> ; XWindowChanges <i>*values</i> ;	Xlib
<b>int XConnectionNumber(<i>display</i>)</b> Display <i>*display</i> ;	Xlib
<b>Constraint</b>	Widgets
<b>void XtConvert(<i>w, from_type, from, to_type, to_return</i>)</b> Widget <i>w</i> ; String <i>from_type, to_type</i> ; XrmValuePtr <i>from, to_return</i> ;	Intrinsics
<b>void XtConvertCase(<i>dpy, keysym, lower_return, upper_return</i>)</b> Display <i>*dpy</i> ; KeySym <i>keysym, *lower_return, *upper_return</i> ;	Intrinsics
<b>XConvertSelection(<i>display, selection, target, property, requestor, time</i>)</b> Display <i>*display</i> ; Atom <i>selection, target, property</i> ; Window <i>requestor</i> ; Time <i>time</i> ;	Xlib
<b>XCopArea(<i>display, src, desc, gc, src_x, src_y, width, height, dest_x, dest_y</i>)</b> Display <i>*display</i> ; Drawable <i>src, dest</i> ; GC <i>gc</i> ; int <i>src_x, src_y, dest_x, dest_y</i> ; unsigned int <i>width, height</i> ;	Xlib
<b>Colormap XCopyColormapAndFree(<i>display, cmap</i>)</b> Display <i>*display</i> ; Colormap <i>cmap</i> ;	Xlib

Function	Manual
<b>XCopyGC</b> ( <i>display, src, valuemask_copy, dest</i> ) Display * <i>display</i> ; GC <i>src, dest</i> ; unsigned int <i>valuemask_copy</i> ; 	Xlib
<b>XCopyPlane</b> ( <i>display, src, dest, gc, src_x, src_y, width, height, dest_x, dest_y, plane</i> ) Display * <i>display</i> ; Drawable <i>src, dest</i> ; GC <i>gc</i> ; int <i>src_x, src_y, dest_x, dest_y</i> ; unsigned int <i>width, height</i> ; unsigned long <i>plane</i> ; 	Xlib
<b>CoreClass</b>	Widgets
<b>void XtCreateApplicationContext()</b>	Intrinsics
<b>Widget XtCreateApplicationShell</b> ( <i>name, widget_class, args, num_args</i> ) String <i>name</i> ; WidgetClass <i>widget_class</i> ; Arglist <i>args</i> ; Cardinal <i>num_args</i> ; 	Intrinsics
<b>XAssocTable *XCreateAssocTable</b> ( <i>size</i> ) int <i>size</i> ; 	Xlib
<b>Pixmap XCreateBitmapFromData</b> ( <i>display, d, data, width, height</i> ) Display * <i>display</i> ; Drawable <i>d</i> ; char * <i>data</i> ; unsigned int <i>width, height</i> ; 	Xlib
<b>Colormap XCreateColormap</b> ( <i>display, w, visual, alloc</i> ) Display * <i>display</i> ; Window <i>w</i> ; Visual * <i>visual</i> ; int <i>alloc</i> ; 	Xlib
<b>Cursor XCreateFontCursor</b> ( <i>display, shape</i> ) Display * <i>display</i> ; unsigned int <i>shape</i> ; 	Xlib



Function	Manual
GC XCreateGC( <i>display</i> , <i>d</i> , <i>valuemask_create</i> , <i>values</i> ) Display <i>*display</i> ; Drawable <i>d</i> ; unsigned long <i>valuemask_create</i> ; XGCValues <i>*values</i> ;	Xlib
Cursor XCreateGlyphCursor( <i>display</i> , <i>source_font</i> , <i>mask_font</i> , <i>source_char</i> , <i>mask_char</i> , <i>foreground_color</i> , <i>background_color</i> ) Display <i>*display</i> ; Font <i>source_font</i> , <i>mask_font</i> ; unsigned int <i>source_char</i> , <i>mask_char</i> ; XColor <i>*foreground_color</i> , <i>*background_color</i> ;	Xlib
XImage *XCreateImage( <i>display</i> , <i>visual</i> , <i>depth</i> , <i>format</i> , <i>offset</i> , <i>data</i> , <i>width</i> , <i>height</i> , <i>bitmap_pat</i> , <i>bytes_per_line</i> ) Display <i>*display</i> ; Visual <i>*visual</i> ; unsigned int <i>depth</i> , <i>width</i> , <i>height</i> ; int <i>format</i> , <i>offset</i> , <i>bitmap_pad</i> , <i>bytes_per_line</i> ; char <i>*data</i> ;	Xlib
Widget XtCreateManagedWidget( <i>name</i> , <i>widget_class</i> , <i>parent</i> , <i>args</i> , <i>num_args</i> ) String <i>name</i> ; WidgetClass <i>widget_class</i> ; Widget <i>parent</i> ; Arglist <i>args</i> ; Cardinal <i>num_args</i> ;	Intrinsics
Pixmap XCreatePixmap( <i>display</i> , <i>d</i> , <i>width</i> , <i>height</i> , <i>depth</i> ) Display <i>*display</i> ; Drawable <i>d</i> ; unsigned int <i>width</i> , <i>height</i> , <i>depth</i> ;	Xlib
Cursor XCreatePixmapCursor( <i>display</i> , <i>source</i> , <i>mask</i> , <i>foreground_color</i> , <i>background_color</i> ) Display <i>*display</i> ; Pixmap <i>source</i> , <i>mask</i> ; XColor <i>*foreground_color</i> , <i>background_color</i> ; unsigned int <i>x</i> , <i>y</i> ;	Xlib



Function	Manual
Pixmap XCreatePixmapFromBitmapData( <i>display, d, data, width, height, fg, bg, depth</i> ) Display <i>*display</i> ; Drawable <i>d</i> ; char <i>*data</i> ; unsigned int <i>width, height, depth</i> ; unsigned long <i>fg, bg</i> ;	Xlib
Widget XtCreatePopupShell( <i>name, widget_class, parent, args, num_args</i> ) String <i>name</i> ; WidgetClass <i>widget_class</i> ; Widget <i>parent</i> ; Arglist <i>args</i> ; Cardinal <i>num_args</i> ;	Intrinsics
Region XCreateRegion()	Xlib
Window XCreateSimpleWindow( <i>display, parent, x, y, width, height, border_width, border, background</i> ) Display <i>*display</i> ; Window <i>parent</i> ; int <i>x, y</i> ; unsigned int <i>width, height, border_width</i> ; unsigned long <i>border, background</i> ;	Xlib
XwCreateTile	Widgets
XtCreateWidget( <i>name, widget_class, parent, args, num_args</i> ) String <i>name</i> ; WidgetClass <i>widget_class</i> ; Widget <i>parent</i> ; Arglist <i>args</i> ; Cardinal <i>num_args</i> ;	Intrinsics
void XtCreateWindow( <i>w, window_class, visual, value_mask, attributes</i> ) Widget <i>w</i> ; unsigned int <i>window_class</i> ; Visual <i>*visual</i> ; XtValueMask <i>value_mask</i> ; XSetWindowAttributes <i>*attributes</i> ;	Intrinsics

Function	Manual
Window XCreateWindow( <i>display</i> , <i>parent</i> , <i>x</i> , <i>y</i> , <i>width</i> , <i>height</i> , <i>border_width</i> , <i>depth</i> , <i>class</i> , <i>visual</i> , <i>valuemask</i> , <i>attributes</i> ) Display * <i>display</i> ; Window <i>parent</i> ; int <i>x</i> , <i>y</i> , <i>depth</i> ; unsigned int <i>width</i> , <i>height</i> , <i>border_width</i> , <i>class</i> ; Visual * <i>visual</i> ; unsigned long <i>valuemask</i> ; XSetWindowAttributes * <i>attributes</i> ;	Xlib
XrmDatabase XtDatabase( <i>display</i> ) Display * <i>display</i> ;	Intrinsics
Colormap XDefaultColormap( <i>display</i> , <i>screen_number</i> ) Display * <i>display</i> ; int <i>screen_number</i> ;	Xlib
Colormap XDefaultColormapOfScreen( <i>screen</i> ) Screen * <i>screen</i> ;	Xlib
int XDefaultDepth( <i>display</i> , <i>screen_number</i> ) Display * <i>display</i> ; int <i>screen_number</i> ;	Xlib
int XDefaultDepthOfScreen( <i>screen</i> ) Screen * <i>screen</i> ;	Xlib
GC XDefaultGC( <i>display</i> , <i>screen_number</i> ) Display * <i>display</i> ; int <i>screen_number</i> ;	Xlib
GC XDefaultGCOfScreen( <i>screen</i> ) Screen * <i>screen</i> ;	Xlib
Window XDefaultRootWindow( <i>display</i> ) Display * <i>display</i> ;	Xlib
int XDefaultScreen( <i>display</i> ) Display * <i>display</i> ;	Xlib
Screen *XDefaultScreenOfDisplay( <i>display</i> ) Display * <i>display</i> ;	Xlib

Function	Manual
Visual *XDefaultVisual( <i>display</i> , <i>screen_number</i> ) Display * <i>display</i> ; int <i>screen_number</i> ;	Xlib
Visual *XDefaultVisualOfScreen( <i>screen</i> ) Screen * <i>screen</i> ;	Xlib
XDefineCursor( <i>display</i> , <i>w</i> , <i>cursor</i> ) Display * <i>display</i> ; Window <i>w</i> ; Cursor <i>cursor</i> ;	Xlib
XDeleteAssoc( <i>display</i> , <i>table</i> , <i>x_id</i> ) Display * <i>display</i> ; XAssocTable * <i>table</i> ; XID <i>x_id</i> ;	Xlib
int XDeleteContext( <i>display</i> , <i>w</i> , <i>context</i> ) Display * <i>display</i> ; Window <i>w</i> ; XContext <i>context</i> ;	Xlib
XModifierKeymap *XDeleteModifiermapEntry( <i>modmap</i> , <i>keycode_entry</i> , <i>modifier</i> ) XModifierKeymap * <i>modmap</i> ; KeyCode <i>keycode_entry</i> ; int <i>modifier</i> ;	Xlib
XDeleteProperty( <i>display</i> , <i>w</i> , <i>property</i> ) Display * <i>display</i> ; Window <i>w</i> ; Atom <i>property</i> ;	Xlib
void XtDestroyApplicationContext( <i>app_context</i> ) XtAppContext <i>app_context</i> ;	Intrinsics
XDestroyAssocTable( <i>table</i> ) XAssocTable * <i>table</i> ;	Xlib
void XtDestroyGC( <i>gc</i> ) GC <i>gc</i> ;	Intrinsics



Function	Manual
int XDestroyImage( <i>ximage</i> ) XImage * <i>ximage</i> ;	Xlib
XDestroyRegion( <i>r</i> ) Region <i>r</i> ;	Xlib
XDestroySubwindows( <i>display</i> , <i>w</i> ) Display * <i>display</i> ; Window <i>w</i> ;	Xlib
void XtDestroyWidget( <i>w</i> ) Widget <i>w</i> ;	Intrinsics
XDestroyWindow( <i>display</i> , <i>w</i> ) Display * <i>display</i> ; Window <i>w</i> ;	Xlib
XHPDeviceAutoRepeatOff( <i>display</i> , <i>deviceid</i> ) Display * <i>display</i> ; XID <i>deviceid</i> ;	Xlib
XHPDeviceAutoRepeatOn( <i>display</i> , <i>deviceid</i> , <i>mode</i> ) Display * <i>display</i> ; XID <i>deviceid</i> ; unsigned int <i>mode</i> ;	Xlib
void XtDirectConvert( <i>converter</i> , <i>args</i> , <i>num_args</i> , <i>from</i> , <i>to_return</i> ) XtConverter <i>converter</i> ; XrmValuePtr <i>args</i> , <i>from</i> , <i>to_return</i> ; Cardinal <i>num_args</i> ;	Intrinsics
XDisableAccessControl( <i>display</i> ) Display * <i>display</i> ;	Xlib
XHPDisableReset( <i>display</i> ) Display * <i>display</i> ;	Xlib
void XtDisownSelection( <i>w</i> , <i>selection</i> , <i>time</i> ) Widget <i>w</i> ; Atom <i>selection</i> ; Time <i>time</i> ;	Intrinsics



Function	Manual
void XtDispatchEvent( <i>event</i> ) XEvent * <i>event</i> ;	Intrinsics
char *XDisplay( <i>string</i> ) char * <i>string</i> ;	Xlib
Display *XtDisplay( <i>w</i> ) Widget <i>w</i> ;	Intrinsics
int XDisplayCells( <i>display</i> , <i>screen_number</i> ) Display * <i>display</i> ; int <i>screen_number</i> ;	Xlib
int XDisplayHeight( <i>display</i> , <i>screen_number</i> ) Display * <i>display</i> ; int <i>screen_number</i> ;	Xlib
int XDisplayHeightMM( <i>display</i> , <i>screen_number</i> ) Display * <i>display</i> ; int <i>screen_number</i> ;	Xlib
void XtDisplayInitialize( <i>app_context</i> , <i>display</i> , <i>application_name</i> , <i>application_class</i> , <i>options</i> , <i>num_options</i> , <i>argc</i> , <i>argv</i> ) XtAppContext <i>app_context</i> ; Display * <i>display</i> ; String <i>application_name</i> , <i>application_class</i> ; XrmOptionDescRec <i>options</i> []; Cardinal <i>num_options</i> , * <i>argc</i> ; String <i>argv</i> [];	Intrinsics
Display *XDisplayOfScreen( <i>screen</i> ) Screen * <i>screen</i> ;	Xlib
int XDisplayPlanes( <i>display</i> , <i>screen_number</i> ) Display * <i>display</i> ; int <i>screen_number</i> ;	Xlib
char *XDisplayString( <i>display</i> ) Display * <i>display</i> ;	Xlib
int XDisplayWidth( <i>display</i> , <i>screen_number</i> ) Display * <i>display</i> ; int <i>screen_number</i> ;	Xlib

Function	Manual
<b>int XDisplayWidthMM(<i>display</i>, <i>screen_number</i>)</b> Display * <i>display</i> ; int <i>screen_number</i> ; 	Xlib
<b>int XDoesBackingStore(<i>screen</i>)</b> Screen * <i>screen</i> ; 	Xlib
<b>Bool XDoesSaveUnders(<i>screen</i>)</b> Screen * <i>screen</i> ; 	Xlib
<b>Status XDraw(<i>display</i>, <i>d</i>, <i>gc</i>, <i>vlist</i>, <i>vcount</i>)</b> Display * <i>display</i> ; Drawable <i>d</i> ; GC <i>gc</i> ; Vertex * <i>vlist</i> ; int <i>vcount</i> ; 	Xlib
<b>XDrawArc(<i>display</i>, <i>d</i>, <i>gc</i>, <i>x</i>, <i>y</i>, <i>width</i>, <i>height</i>, <i>angle1</i>, <i>angle2</i>)</b> Display * <i>display</i> ; Drawable <i>d</i> ; GC <i>gc</i> ; int <i>x</i> , <i>y</i> , <i>angle1</i> , <i>angle2</i> ; unsigned int <i>width</i> , <i>height</i> ; 	Xlib
<b>XDrawArcs(<i>display</i>, <i>d</i>, <i>gc</i>, <i>arcs</i>, <i>narcs</i>)</b> Display * <i>display</i> ; Drawable <i>d</i> ; GC <i>gc</i> ; XArc * <i>arcs</i> ; int <i>narcs</i> ; 	Xlib
<b>Status XDrawFilled(<i>display</i>, <i>d</i>, <i>gc</i>, <i>vlist</i>, <i>vcount</i>)</b> Display * <i>display</i> ; Drawable <i>d</i> ; GC <i>gc</i> ; Vertex * <i>vlist</i> ; int <i>vcount</i> ; 	Xlib

Function	Manual
<hr/> XDrawImageString( <i>display, d, gc, x, y, string, length</i> ) Display * <i>display</i> ; Drawable <i>d</i> ; GC <i>gc</i> ; int <i>x, y, length</i> ; char * <i>string</i> ; <hr/>	Xlib
XDrawImageString16( <i>display, d, gc, x, y, string, length</i> ) Display * <i>display</i> ; Drawable <i>d</i> ; GC <i>gc</i> ; int <i>x, y, length</i> ; XChar2b * <i>string</i> ; <hr/>	Xlib
XDrawLine( <i>display, d, gc, x1, y1, x2, y2</i> ) Display * <i>display</i> ; Drawable <i>d</i> ; GC <i>gc</i> ; int <i>x1, y1, x2, y2</i> ; <hr/>	Xlib
XDrawLines( <i>display, d, gc, points, npoints, mode</i> ) Display * <i>display</i> ; Drawable <i>d</i> ; GC <i>gc</i> ; XPoint * <i>points</i> ; int <i>npoints, mode</i> ; <hr/>	Xlib
XDrawPoint( <i>display, d, gc, x, y</i> ) Display * <i>display</i> ; Drawable <i>d</i> ; GC <i>gc</i> ; int <i>x, y</i> ; <hr/>	Xlib
XDrawPoints( <i>display, d, gc, points, npoints, mode</i> ) Display * <i>display</i> ; Drawable <i>d</i> ; GC <i>gc</i> ; XPoint * <i>points</i> ; int <i>npoints, mode</i> ; <hr/>	Xlib

Function	Manual
<hr/> XDrawRectangle( <i>display, d, g, x, y, width, height</i> ) Display * <i>display</i> ; Drawable <i>d</i> ; GC <i>gc</i> ; int <i>x, y</i> ; unsigned int <i>width, height</i> ; <hr/>	Xlib
XDrawRectangles( <i>display, d, gc, rectangles, nrectangles</i> ) Display * <i>display</i> ; Drawable <i>d</i> ; GC <i>gc</i> ; XRectangle <i>rectangles</i> []; int <i>nrectangles</i> ; <hr/>	Xlib
XDrawSegments( <i>display, d, gc, segments, nsegments</i> ) Display * <i>display</i> ; Drawable <i>d</i> ; GC <i>gc</i> ; XSegment * <i>segments</i> ; int <i>nsegments</i> ; <hr/>	Xlib
XDrawString( <i>display, d, gc, x, y, string, length</i> ) Display * <i>display</i> ; Drawable <i>d</i> ; GC <i>gc</i> ; int <i>x, y, length</i> ; char * <i>string</i> ; <hr/>	Xlib
XDrawString16( <i>display, d, gc, x, y, string, length</i> ) Display * <i>display</i> ; Drawable <i>d</i> ; GC <i>gc</i> ; int <i>x, y, length</i> ; XChar2b * <i>string</i> ; <hr/>	Xlib
XDrawText( <i>display, d, gc, x, y, items, nitems</i> ) Display * <i>display</i> ; Drawable <i>d</i> ; GC <i>gc</i> ; int <i>x, y, nitems</i> ; XTextItem * <i>items</i> ; <hr/>	Xlib



Function	Manual
<b>XDrawText16</b> ( <i>display, d, gc, x, y, items, nitems</i> ) Display <i>*display</i> ; Drawable <i>d</i> ; GC <i>gc</i> ; int <i>x, y, nitems</i> ; XTextItem16 <i>*items</i> ;	Xlib
<b>Bool XEmptyRegion</b> ( <i>r</i> ) Region <i>r</i> ;	Xlib
<b>XEnableAccessControl</b> ( <i>display</i> ) Display <i>*display</i> ;	Xlib
<b>XHPEnableReset</b> ( <i>display</i> ) Display <i>*display</i> ;	Xlib
<b>Bool XEqualRegion</b> ( <i>r1, r2</i> ) Region <i>r1, r2</i> ;	Xlib
<b>void XtError</b> ( <i>message</i> ) String <i>message</i> ;	Intrinsics
<b>void XtErrorMsg</b> ( <i>name, type, class, defaultp, params, num_params</i> ) String <i>name, type, class, defaultp, *params</i> ; Cardinal <i>num_params</i> ;	Intrinsics
<b>long XEventMaskOfScreen</b> ( <i>screen</i> ) Screen <i>*screen</i> ;	Xlib
<b>int XEventQueued</b> ( <i>display, mode</i> ) Display <i>*display</i> ; int <i>mode</i> ;	Xlib
<b>char *XFetchBuffer</b> ( <i>display, nbytes_return, buffer</i> ) Display <i>*display</i> ; int <i>*nbytes_return, buffer</i> ;	Xlib
<b>char *XFetchBytes</b> ( <i>display, nbytes_return</i> ) Display <i>*display</i> ; int <i>*nbytes_return</i> ;	Xlib
<b>Status XFetchName</b> ( <i>display, w, window_name_return</i> ) Display <i>*display</i> ; Window <i>w</i> ; char <i>**window_name_return</i> ;	Xlib

Function	Manual
<pre>int XHPFileToPixmap(<i>display, pixmap, cmap, gc, src_x, src_y, dst_x, dst_y, width, height, filename</i>)     Display *<i>display</i>;     Pixmap <i>pixmap</i>;     Colormap <i>cmap</i>;     GC <i>gc</i>;     int <i>src_x, src_y, dst_x, dst_y</i>;     unsigned int <i>width, height</i>;     char <i>filename</i>;</pre>	Xlib
<pre>int XHPFileToWindow(<i>display, w, modify_cmap, gc, src_x, src_y, dst_x, dst_y, width, height, filename</i>)     Display *<i>display</i>;     Window <i>w</i>;     int <i>modify_cmap, src_x, src_y, dst_x, dst_y</i>;     GC <i>gc</i>;     unsigned int <i>width, height</i>;     char <i>filename</i>;</pre>	Xlib
<pre>XFillArc(<i>display, d, gc, x, y, width, height, angle1, angle2</i>)     Display *<i>display</i>;     Drawable <i>d</i>;     GC <i>gc</i>;     int <i>x, y, angle1, angle2</i>;     unsigned int <i>width, height</i>;</pre>	Xlib
<pre>XFillArcs(<i>display, d, gc, arcs, narcs</i>)     Display *<i>display</i>;     Drawable <i>d</i>;     GC <i>gc</i>;     XArc *<i>arcs</i>;     int <i>narcs</i>;</pre>	Xlib

Function	Manual
XFillPolygon( <i>display, d, gc, points, npoints, shape, mode</i> ) Display * <i>display</i> ; Drawable <i>d</i> ; GC <i>gc</i> ; XPoint * <i>points</i> ; int <i>npoints, shape, mode</i> ;	Xlib
XFillRectangle( <i>display, d, gc, x, y, width, height</i> ) Display * <i>display</i> ; Drawable <i>d</i> ; GC <i>gc</i> ; int <i>x, y</i> ; unsigned int <i>width, height</i> ;	Xlib
XFillRectangles( <i>display, d, gc, rectangles, nrectangles</i> ) Display * <i>display</i> ; Drawable <i>d</i> ; GC <i>gc</i> ; XRectangle * <i>rectangles</i> ; int <i>nrectangles</i> ;	Xlib
int XFindContext( <i>display, w, context, data_return</i> ) Display * <i>display</i> ; Window <i>w</i> ; XContext <i>context</i> ; caddr_t * <i>data_return</i> ;	Xlib
XExtData *XFindOnExtensionList( <i>structure, number</i> ) struct _XExtData ** <i>structure</i> ; int <i>number</i> ;	Xlib
XFlush( <i>display</i> ) Display * <i>display</i> ;	Xlib
FlushGC( <i>display, gc</i> ) Display * <i>display</i> ; GC <i>gc</i> ;	Xlib
XwformWidgetClass	Widgets



Function	Manual
<code>XForceScreenSaver(<i>display</i>, <i>mode</i>)</code> <code>Display *<i>display</i>;</code> <code>int <i>mode</i>;</code>	Xlib
<code>XwframeWidgetClass</code>	Widgets
<code>XFree(<i>data</i>)</code> <code>char *<i>data</i>;</code>	Xlib
<code>void XtFree(<i>ptr</i>)</code> <code>char *<i>ptr</i>;</code>	Intrinsics
<code>XFreeColormap(<i>display</i>, <i>cmap</i>)</code> <code>Display *<i>display</i>;</code> <code>Colormap <i>cmap</i>;</code>	Xlib
<code>XFreeColors(<i>display</i>, <i>cmap</i>, <i>pixels</i>, <i>npixels</i>, <i>planes</i>)</code> <code>Display *<i>display</i>;</code> <code>Colormap <i>cmap</i>;</code> <code>unsigned long <i>pixels</i>[], <i>planes</i>;</code> <code>int <i>npixels</i>;</code>	Xlib
<code>XFreeCursor(<i>display</i>, <i>cursor</i>)</code> <code>Display *<i>display</i>;</code> <code>Cursor <i>cursor</i>;</code>	Xlib
<code>void XHPFreeDeviceList(<i>list</i>)</code> <code>XHPDeviceList *<i>list</i>;</code>	Xlib
<code>XFreeFont(<i>display</i>, <i>font_struct</i>)</code> <code>Display *<i>display</i>;</code> <code>XFontStruct *<i>font_struct</i>;</code>	Xlib
<code>XFreeFontInfo(<i>names</i>, <i>free_info</i>, <i>actual_count</i>)</code> <code>char **<i>names</i>;</code> <code>XFontStruct *<i>free_info</i>;</code> <code>int <i>actual_count</i>;</code>	Xlib
<code>XFreeFontName(<i>list</i>)</code> <code>char *<i>list</i>[];</code>	Xlib
<code>XFreeFontPath(<i>list</i>)</code> <code>char **<i>list</i>;</code>	Xlib

Function	Manual
<code>XFreeGC(<i>display</i>, <i>gc</i>)</code> Display <i>*display</i> ; GC <i>gc</i> ;	Xlib
<code>XFreeModifiermap(<i>modmap</i>)</code> XModifierKeymap <i>*modmap</i> ;	Xlib
<code>XFreePixmap(<i>display</i>, <i>pixmap</i>)</code> Display <i>*display</i> ; Pixmap <i>pixmap</i> ;	Xlib
<code>GContext XGContextFromGC(<i>gc</i>)</code> GC <i>gc</i> ;	Xlib
<code>int XGeometry(<i>display</i>, <i>screen</i>, <i>position</i>, <i>default_position</i>, <i>bwidth</i>, <i>fwidth</i>, <i>fheight</i>, <i>xadder</i>, <i>yadder</i>, <i>x_return</i>, <i>y_return</i>, <i>width_return</i>, <i>height_return</i>)</code> Display <i>*display</i> ; int <i>screen</i> , <i>xadder</i> , <i>yadder</i> , <i>x_return</i> , <i>y_return</i> ; char <i>*position</i> , <i>*default_position</i> ; unsigned int <i>bwidth</i> , <i>fwidth</i> , <i>fheight</i> ; int <i>*width_return</i> , <i>*height_return</i> ;	Xlib
<code>XFontStruct *XHPGet16bitMixedFont(<i>font</i>)</code> XFontStruct <i>font</i> ;	Xlib
<code>void XtGetApplicationResources(<i>w</i>, <i>base</i>, <i>resources</i>, <i>num_resources</i>, <i>args</i>, <i>num_args</i>)</code> Widget <i>w</i> ; caddr_t <i>base</i> ; XtResourceList <i>resources</i> ; Cardinal <i>num_resources</i> , <i>num_args</i> ; ArgList <i>args</i> ;	Intrinsics
<code>char *XGetAtomName(<i>display</i>, <i>atom</i>)</code> Display <i>*display</i> ; Atom <i>atom</i> ;	Xlib
<code>Status XGetClassHint(<i>display</i>, <i>w</i>, <i>class_hints_return</i>)</code> Display <i>*display</i> ; Window <i>w</i> ; XClassHint <i>*class_hints_return</i> ;	Xlib

Function	Manual
XHPGetCurrentDeviceMask( <i>display, window, deviceid, mask_return</i> ) Display * <i>display</i> ; Window <i>window</i> ; XID <i>deviceid</i> ; Mask * <i>mask_return</i> ;	Xlib
char *XGetDefault( <i>display, program, options</i> ) Display * <i>display</i> ; char * <i>program, *options</i> ;	Xlib
XHPGetDeviceControl( <i>display, deviceid, values_return</i> ) Display * <i>display</i> ; XID <i>deviceid</i> ; XHPDeviceState * <i>values_return</i> ;	Xlib
XHPGetDeviceFocus( <i>display, deviceid, focus_return, revert_to_return</i> ) Display * <i>display</i> ; XID <i>deviceid</i> ; Window * <i>focus_return</i> ; int * <i>revert_to_return</i> ;	Xlib
KeySym *XHPGetDeviceKeyMapping( <i>display, deviceid, first_keycode_wanted, keycode_count, keysyms_per_keycode_return</i> ) Display * <i>display</i> ; XID <i>deviceid</i> ; KeyCode <i>first_keycode_wanted</i> ; int <i>keycode_count, *keysyms_per_keycode_return</i> ;	Xlib
XModifierKeyMap *XHPGetDevideModifierMapping( <i>display, deviceid</i> ) Display * <i>display</i> ; XID <i>deviceid</i> ;	Xlib
XHPTimeCoord *XHPGetDeviceMotionEvents( <i>display, w, deviceid, start, stop, nevents_return</i> ) Display * <i>display</i> ; Window <i>w</i> ; XID <i>deviceid</i> ; Time <i>start, stop</i> ; int * <i>nevents_return</i> ;	Xlib









Function	Manual
<b>XGetErrorDatabaseText</b> ( <i>display, name, message, default_string, buffer_return, length</i> ) Display *display; char *name, *message, *default_string; char *buffer_return; int length;	Xlib
<b>XGetErrorText</b> ( <i>display, code, buffer_return, length</i> ) Display *display; int code, length; char *buffer_return;	Xlib
<b>int XHPGetExtEventMask</b> ( <i>display, event_constant, event_type, mask</i> ) Display *display; long event_constant, *event_type, *mask;	Xlib
<b>XrmDatabase XrmGetFileDatabase</b> ( <i>filename</i> ) char *filename;	Xlib
<b>char **XGetFontPath</b> ( <i>display, npaths_return</i> ) Display *display; int *npaths_return;	Xlib
<b>Bool XGetFontProperty</b> ( <i>font_struct, atom, value_return</i> ) XFontStruct *font_struct; Atom atom; unsigned long *value_return;	Xlib
<b>GC XtGetGC</b> ( <i>w, value_mask, values</i> ) Widget w; XtGCMask value_mask; XGCValues *values;	Intrinsics
<b>Status XGetGeometry</b> ( <i>display, d, root_return, x_return, y_return, width_return, height_return, border_width_return, depth_return</i> ) Display *display; Drawable d; Window *root_return; int *x_return, *y_return; unsigned int *width_return, *height_return; unsigned int *border_width_return, *depth_return;	Xlib

Function	Manual
<b>Status</b> <i>XGetIconName</i> ( <i>display</i> , <i>w</i> , <i>icon_name_return</i> ) Display * <i>display</i> ; Window <i>w</i> ; char ** <i>icon_name_return</i> ; 	Xlib
<b>Status</b> <i>XGetIconSizes</i> ( <i>display</i> , <i>w</i> , <i>size_list_return</i> , <i>count_return</i> ) Display * <i>display</i> ; Window <i>w</i> ; XIconSize ** <i>size_list_return</i> ; int * <i>count_return</i> ; 	Xlib
<b>XImage</b> * <i>XGetImage</i> ( <i>display</i> , <i>d</i> , <i>x</i> , <i>y</i> , <i>width</i> , <i>height</i> , <i>plane_mask</i> , <i>format</i> ) Display * <i>display</i> ; Drawable <i>d</i> ; int <i>x</i> , <i>y</i> , <i>format</i> ; unsigned int <i>width</i> , <i>height</i> ; long <i>plane_mask</i> ; 	Xlib
<b>XGetInputFocus</b> ( <i>display</i> , <i>focus_return</i> , <i>revert_to_return</i> ) Display * <i>display</i> ; Window * <i>focus_return</i> ; int * <i>revert_to_return</i> ; 	Xlib
<b>XGetKeyboardControl</b> ( <i>display</i> , <i>values_return</i> ) Display * <i>display</i> ; XKeyboardState * <i>values_return</i> ; 	Xlib
<b>KeySym</b> * <i>XGetKeyboardMapping</i> ( <i>display</i> , <i>first_keycode_wanted</i> , <i>keycode_count</i> , <i>keysyms_per_keycode_return</i> ) Display * <i>display</i> ; KeyCode <i>first_keycode_wanted</i> ; int <i>keycode_count</i> , * <i>keysyms_per_keycode_return</i> ; 	Xlib
<b>XModifierKeymap</b> * <i>XGetModifiersMapping</i> ( <i>display</i> ) Display * <i>display</i> ; 	Xlib

Function	Manual
<b>XTimeCoord *XGetMotionEvents</b> ( <i>display</i> , <i>w</i> , <i>start</i> , <i>stop</i> , <i>nevents_return</i> ) Display * <i>display</i> ; Window <i>w</i> ; Time <i>start</i> , <i>stop</i> ; int * <i>nevents_return</i> ;	Xlib
<b>Status XGetNormalHints</b> ( <i>display</i> , <i>w</i> , <i>hints_return</i> ) Display * <i>display</i> ; Window <i>w</i> ; XSizeHints * <i>hints_return</i> ;	Xlib
<b>unsigned long XGetPixel</b> ( <i>ximage</i> , <i>x</i> , <i>y</i> ) XImage * <i>ximage</i> ; int <i>x</i> , <i>y</i> ;	Xlib
<b>XGetPointerControl</b> ( <i>display</i> , <i>accel_numerator_return</i> , <i>accel_denominator_return</i> , <i>threshold_return</i> ) Display * <i>display</i> ; int <i>accel_numerator_return</i> , * <i>accel_denominator_return</i> ; int * <i>threshold_return</i> ;	Xlib
<b>int XGetPointerMapping</b> ( <i>display</i> , <i>map_return</i> , <i>nmap</i> ) Display * <i>display</i> ; unsigned char <i>map_return</i> [ ]; int <i>nmap</i> ;	Xlib
<b>XGetPointerParameters</b> ( <i>display</i> , <i>accel_numerator_return</i> , <i>accel_denominator_return</i> , <i>threshold_return</i> ) Display * <i>display</i> ; int * <i>accel_numerator_return</i> , * <i>accel_denominator_return</i> ; int * <i>threshold_return</i> ;	Xlib
<b>Bool XrmGetResource</b> ( <i>database</i> , <i>str_name</i> , <i>str_class</i> , <i>str_type_return</i> , <i>str_value_return</i> ) XrmDatabase <i>database</i> ; char * <i>str_name</i> , * <i>str_class</i> , ** <i>str_type_return</i> ; XrmValue * <i>value_return</i> ;	Xlib
<b>void XtGetResourceList</b> ( <i>class</i> , <i>resources_return</i> , <i>num_resources_return</i> ) WidgetClass <i>class</i> ; XtResourceList * <i>resources_return</i> ; Cardinal * <i>num_resources_return</i> ;	Intrinsics



Function	Manual
<hr/> XGetScreenSaver( <i>display</i> , <i>timeout_return</i> , <i>interval_return</i> , <i>prefer_blanking_return</i> , <i>allow_exposures_return</i> ) Display * <i>display</i> ; int * <i>timeout_return</i> , * <i>interval_return</i> ; int * <i>prefer_blanking_return</i> , * <i>allow_exposures_return</i> ; <hr/>	Xlib
Window XGetSelectionOwner( <i>display</i> , <i>selection</i> ) Display * <i>display</i> ; Atom <i>selection</i> ; <hr/>	Xlib
void XtGetSelectionValue( <i>w</i> , <i>selection</i> , <i>target</i> , <i>callback</i> , <i>client_data</i> , <i>time</i> ); Widget <i>w</i> ; Atom <i>selection</i> , <i>target</i> ; XtSelectionCallbackProc <i>callback</i> ; caddr_t <i>client_data</i> ; Time <i>time</i> ; <hr/>	Intrinsics
XHPGetServerMode( <i>display</i> , <i>screen</i> ) Display * <i>display</i> ; int <i>screen</i> ; <hr/>	Xlib
Status XGetSizeHints( <i>display</i> , <i>w</i> , <i>hints_return</i> , <i>property</i> ) Display * <i>display</i> ; Window <i>w</i> ; XSizeHints * <i>hints_return</i> ; Atom <i>property</i> ; <hr/>	Xlib
Status XGetStandardColormap( <i>display</i> , <i>w</i> , <i>cmap_return</i> , <i>property</i> ) Display * <i>display</i> ; Window <i>w</i> ; XStandardColormap * <i>cmap_return</i> ; Atom <i>property</i> ; <hr/>	Xlib
XrmDatabase XrmGetStringDatabase( <i>string</i> ) char * <i>data</i> ; <hr/>	Xlib

Function	Manual
<hr/>  <pre> XImage *XGetSubImage(<i>display, d, x, y, width, height, plane_mask, format, dest_image, dest_x, dest_y</i>)     Display *<i>display</i>;     Drawable <i>d</i>;     int <i>x, y, format, dest_x, dest_y</i>;     unsigned int <i>width, height</i>;     unsigned long <i>plane_mask</i>;     XImage *<i>dest_image</i>; </pre>	Xlib
<hr/>  <pre> void XtGetSubresources(<i>w, base, name, class, resources, num_resources, args, num_args</i>)     Widget <i>w</i>;     caddr_t <i>base</i>;     String <i>name, class</i>;     XtResourceList <i>resources</i>;     Cardinal <i>num_resources, num_args</i>;     ArgList <i>args</i>; </pre>	Intrinsics
<hr/>  <pre> void XtGetSubvalues(<i>base, resources, num_resources, args, num_args</i>)     caddr_t <i>base</i>;     XtResourceList <i>resource</i>;     Cardinal <i>num_resources, num_args</i>;     ArgList <i>args</i>; </pre>	Intrinsics
<hr/>  <pre> Status XGetTransientForHint(<i>display, w, prop_window_return</i>)     Display *<i>display</i>;     Window <i>w</i>;     Window *<i>prop_window_return</i>; </pre>	Xlib
<hr/>  <pre> void XtGetValues(<i>w, args, num_args</i>)     Widget <i>w</i>;     ArgList <i>args</i>;     Cardinal <i>num_args</i>; </pre>	Intrinsics
<hr/>  <pre> XVisualInfo *XGetVisualInfo(<i>display, vinfo_mask, vinfo_template, nitems_return</i>)     Display *<i>display</i>;     long <i>vinfo_mask</i>;     XVisualInfo *<i>vinfo_template</i>;     int *<i>nitems_return</i>; </pre>	Xlib

Function	Manual
Status XGetWindowAttributes( <i>display, w, window_attributes_return</i> ) Display * <i>display</i> ; Window <i>w</i> ; XWindowAttributes * <i>window_attributes_return</i> ;	Xlib
int XGetWindowProperty( <i>display, w, property, long_offset, long_length, delete, req_type, actual_type_return, actual_format_return, nitems_return, bytes_after_return, prop_return</i> ) Display * <i>display</i> ; Window <i>w</i> ; Atom <i>property, req_type, *actual_type_return</i> ; long <i>long_offset, long_length</i> ; Bool <i>delete</i> ; int * <i>actual_format_return</i> ; unsigned long * <i>nitem_return, *bytes_after_return, **prop_return</i> ;	Xlib
XWMHints *XGetWMHints( <i>display, w</i> ) Display * <i>display</i> ; Window <i>w</i> ;	Xlib
Status XGetZoomHints( <i>display, w, zhints_return</i> ) Display * <i>display</i> ; Window <i>w</i> ; XSizeHints * <i>zhints_return</i> ;	Xlib
XGrabButton( <i>display, button_grab, modifiers, grab_window, owner_events, event_mask, pointer_mode, keyboard_mode, confine_to, cursor</i> ) Display * <i>display</i> ; unsigned int <i>button_grab, modifiers, event_mask</i> ; Window <i>grab_window, confine_to</i> ; Bool <i>owner_events</i> ; int <i>pointer_mode, keyboard_mode</i> ; Cursor <i>cursor</i> ;	Xlib
XHPGrabDevice( <i>display, deviceid, grab_window, pointer_mode, device_mode, owner_events, time</i> ) Display * <i>display</i> ; char <i>deviceid</i> ; Window <i>grab_window</i> ; int <i>pointer_mode, device_mode</i> ; Bool <i>owner_events</i> ; Time <i>time</i> ;	Xlib



Function	Manual
<hr/> XHPGrabDeviceButton( <i>display, deviceid, button, modifiers, grab_window, owner_events, event_mask, pointer_mode, device_mode</i> ) Display *display; XID deviceid; unsigned int button, modifiers, event_mask; Window grab_window; Bool owner_events; int pointer_mode, device_mode; <hr/>	Xlib
XHPGrabDeviceKey( <i>display, deviceid, keycode, modifiers, grab_window, owner_events, pointer_mode, device_mode</i> ) Display *display; XID deviceid; unsigned int keycode, modifiers; Window grab_window; Bool owner_events; int pointer_mode, device_mode; <hr/>	Xlib
XGrabKey( <i>display, keycode, modifiers, grab_window, owner_events, pointer_mode, keyboard_mode</i> ) Display *display; int keycode, pointer_mode, keyboard_mode; unsigned int modifiers; Window grab_window; Bool owner_events; <hr/>	Xlib
int XGrabKeyboard( <i>display, grab_window, owner_events, pointer_mode, keyboard_mode, time</i> ) Display *display; Window grab_window; Bool owner_events; int pointer_mode, keyboard_mode; Time time; <hr/>	Xlib

Function	Manual
<b>int XGrabPointer</b> ( <i>display, grab_window, owner_events, event_mask, pointer_mode, keyboard_mode, confine_to, cursor, time</i> ) Display *display; Window grab_window, confine_to; Bool owner_events; unsigned int event_masks; int pointer_mode, keyboard_mode; Cursor cursor; Time time;	Xlib
<b>XGrabServer</b> ( <i>display</i> ) Display *display;	Xlib
<b>XtCallbackStatus XtHasCallbacks</b> ( <i>w, callback_name</i> ) Widget w; String callback_name;	Intrinsics
<b>int XHeightOfScreen</b> ( <i>screen</i> ) Screen *screen;	Xlib
<b>int XHeightMMOfScreen</b> ( <i>screen</i> ) Screen *screen;	Xlib
<b>XIfEvent</b> ( <i>display, event_return, predicate, arg</i> ) Display *display; XEvent *event_return; Bool (*predicate)(); char *arg;	Xlib
<b>int XImageByteOrder</b> ( <i>display</i> ) Display *display;	Xlib
<b>XwimageEditWidgetClass</b>	Widgets
<b>XExtCodes *XInitExtension</b> ( <i>display, name</i> ) Display *display; char *name;	Xlib
<b>void XrmInitialize</b> ()	Xlib

Function	Manual
Widget XtInitialize( <i>shell_name</i> , <i>application_class</i> , <i>options</i> , <i>num_options</i> , <i>argc</i> , <i>argv</i> ) String <i>shell_name</i> , <i>application_class</i> ; XrmOptionDescRec <i>options</i> [ <i>  </i> ]; Cardinal <i>num_options</i> , <i>*argc</i> ; String <i>argv</i> [ <i>  </i> ];	Intrinsics
void InsertChild( <i>w</i> ) Widget <i>w</i> ;	Intrinsics
XModifierKeymap *XInsertModifiermapEntry( <i>modmap</i> , <i>keycode_entry</i> , <i>modifier</i> ) XModifierKeymap <i>*modmap</i> ; KeyCode <i>keycode_entry</i> ; int <i>modifier</i> ;	Xlib
void XtInstallAccelerators( <i>destination</i> , <i>source</i> ) Widget <i>destination</i> , <i>source</i> ;	Intrinsics
void XtInstallAllAccelerators( <i>destination</i> , <i>source</i> ) Widget <i>destination</i> , <i>source</i> ;	Intrinsics
XInstallColormap( <i>display</i> , <i>cmap</i> ) Display <i>*display</i> ; Colormap <i>cmap</i> ;	Xlib
Atom XInternAtom( <i>display</i> , <i>atom_name</i> , <i>only_if_exists</i> ) Display <i>*display</i> ; char <i>*atom_name</i> ; Bool <i>only_if_exists</i> ;	Xlib
XIntersectRegion( <i>sra</i> , <i>srb</i> , <i>dr_return</i> ) Region <i>sra</i> , <i>srb</i> , <i>dr_return</i> ;	Xlib
Bool XHPis16bitCharacter( <i>font</i> , <i>byte1</i> , <i>byte2</i> ) Font <i>font</i> ; unsigned char <i>byte1</i> , <i>byte2</i> ;	Xlib
void XtIsComposite( <i>w</i> ) Widget <i>w</i> ;	Intrinsics
IsCursorKey( <i>keysym</i> ) KeySym <i>keysym</i> ;	Xlib



Function	Manual
<b>IsFunctionKey</b> ( <i>keysym</i> ) KeySym <i>keysym</i> ;	Xlib
<b>IsKeypadKey</b> ( <i>keysym</i> ) KeySym <i>keysym</i> ;	Xlib
<b>IsMiscFunctionKey</b> ( <i>keysym</i> ) KeySym <i>keysym</i> ;	Xlib
<b>IsModifierKey</b> ( <i>keysym</i> ) KeySym <i>keysym</i> ;	Xlib
<b>IsPFKey</b> ( <i>keysym</i> ) KeySym <i>keysym</i> ;	Xlib
<b>Boolean XtIsRealized</b> ( <i>w</i> ) Widget <i>w</i> ;	Intrinsics
<b>Boolean XtIsSensitive</b> ( <i>w</i> ) Widget <i>w</i> ;	Intrinsics
<b>Boolean XtIsSubclass</b> ( <i>w</i> , <i>widget_class</i> ) Widget <i>w</i> ; WidgetClass <i>widget_class</i> ;	Intrinsics
<b>KeySym XKeycodeToKeysym</b> ( <i>display</i> , <i>keycode</i> , <i>index</i> ) Display <i>*display</i> ; KeyCode <i>keycode</i> ; int <i>index</i> ;	Xlib
<b>KeyCode XKeysymToKeycode</b> ( <i>display</i> , <i>keysym</i> ) Display <i>*display</i> ; KeySym <i>keysym</i> ;	Xlib
<b>int XHPKeysymToRoman8</b> ( <i>keysym</i> , <i>r8_return</i> ) Keysym <i>keysym</i> ; char <i>*r8_return</i> ;	Xlib
<b>char *XKeysymToString</b> ( <i>keysym</i> ) KeySym <i>keysym</i> ;	Xlib
<b>XKillClient</b> ( <i>display</i> , <i>resource</i> ) Display <i>*display</i> ; XID <i>resource</i> ;	Xlib

Function	Manual
int XLastKnownRequestProcessed( <i>display</i> ) Display * <i>display</i> ;	Xlib
char **XListExtensions( <i>display</i> , <i>nextensions_return</i> ) Display * <i>display</i> ; int * <i>nextentions_return</i> ;	Xlib
char **XListFonts( <i>display</i> , <i>pattern</i> , <i>maxnames</i> , <i>actual_count_return</i> ) Display * <i>display</i> ; char * <i>pattern</i> ; int <i>maxnames</i> , * <i>actual_count_return</i> ;	Xlib
char **XListFontsWithInfo( <i>display</i> , <i>pattern</i> , <i>maxnames</i> , <i>count_return</i> , <i>info_return</i> ) Display * <i>display</i> ; char * <i>pattern</i> ; int <i>maxnames</i> , * <i>count_return</i> ; XFontStruct ** <i>info_return</i> ;	Xlib
XHostAddress *XListHosts( <i>display</i> , <i>nhosts_return</i> , <i>state_return</i> ) Display * <i>display</i> ; int * <i>nhosts_return</i> ; Bool * <i>state_return</i> ;	Xlib
XHPDeviceList *XHPListInputDevices( <i>display</i> , <i>ndevices</i> ) Display * <i>display</i> ; int * <i>ndevices</i> ;	Xlib
Colormap *XListInstalledColormaps( <i>display</i> , <i>w</i> , <i>num_return</i> ) Display * <i>display</i> ; Window <i>w</i> ; int * <i>num_return</i> ;	Xlib
Atom *XListProperties( <i>display</i> , <i>w</i> , <i>num_prop_return</i> ) Display * <i>display</i> ; Window <i>w</i> ; int * <i>num_prop_returned</i> ;	Xlib
XwlistWidgetClass	Widgets
Font XLoadFont( <i>display</i> , <i>name</i> ) Display * <i>display</i> ; char * <i>name</i> ;	Xlib

Function	Manual
<b>XFontStruct *XLoadQueryFont</b> ( <i>display</i> , <i>name</i> ) Display * <i>display</i> ; char * <i>name</i> ;	Xlib
<b>LockDisplay</b> ( <i>display</i> ) Display * <i>display</i> ;	Xlib
<b>char *XLookupAssoc</b> ( <i>display</i> , <i>table</i> , <i>x_id</i> ) Display * <i>display</i> ; XAssocTable * <i>table</i> ; XID <i>x_id</i> ;	Xlib
<b>Status XLookupColor</b> ( <i>display</i> , <i>cmap</i> , <i>color_name</i> , <i>exact_def_return</i> , <i>screen_def_return</i> ) Display * <i>display</i> ; Colormap <i>cmap</i> ; char * <i>color_name</i> ; XColor * <i>exact_def_return</i> , * <i>screen_def_return</i> ;	Xlib
<b>KeySym XLookupKeysym</b> ( <i>key_event</i> , <i>index</i> ) XKeyEvent * <i>key_event</i> ; int <i>index</i> ;	Xlib
<b>int XLookupString</b> ( <i>event_struct</i> , <i>buffer_return</i> , <i>bytes_buffer</i> , <i>keysym_return</i> , <i>status_in_out</i> ) XKeyEvent * <i>event_struct</i> ; char * <i>buffer_return</i> ; int <i>bytes_buffer</i> ; KeySym * <i>keysym_return</i> ; XComposeStatus * <i>status_in_out</i> ;	Xlib
<b>XLowerWindow</b> ( <i>display</i> , <i>w</i> ) Display * <i>display</i> ; Window <i>w</i> ;	Xlib
<b>void XtMainLoop</b> ()	Intrinsics
<b>XMakeAssoc</b> ( <i>display</i> , <i>table</i> , <i>x_id</i> , <i>data</i> ) Display * <i>display</i> ; XAssocTable * <i>table</i> ; XID <i>x_id</i> ; char * <i>data</i> ;	Xlib



Function	Manual
XtGeometryResult XtMakeGeometryRequest( <i>w, request, reply_return</i> ) Widget <i>w</i> ; XtWidgetGeometry * <i>request, reply_return</i> ;	Intrinsics
XtGeometryResult XtMakeResizeRequest( <i>w, width, height, width_return, height_return</i> ) Widget <i>w</i> ; Dimension <i>width, height</i> ; Dimension * <i>width_return, *height_return</i> ;	Intrinsics
char *XtMalloc( <i>size</i> ) Cardinal <i>size</i> ;	Intrinsics
void XtManageChild( <i>child</i> ) Widget <i>child</i> ;	Intrinsics
void XtManageChildren( <i>children, num_children</i> ) WidgetList <i>children</i> ; Cardinal <i>num_children</i> ;	Intrinsics
XwmanagerWidgetClass	Widgets
XMapRaised( <i>display, w</i> ) Display * <i>display</i> ; Window <i>w</i> ;	Xlib
XMapSubwindows( <i>display, w</i> ) Display * <i>display</i> ; Window <i>w</i> ;	Xlib
XtMapWidget( <i>w</i> ) Widget <i>w</i> ;	Intrinsics
XMapWindow( <i>display, w</i> ) Display * <i>display</i> ; Window <i>w</i> ;	Xlib
XMaskEvent( <i>display, event_mask, event_return</i> ) Display * <i>display</i> ; long <i>event_mask</i> ; XEvent * <i>event_return</i> ;	Xlib

Function	Manual
Status XMatchVisualInfo( <i>display, screen, depth, class, vinfo_return</i> ) Display * <i>display</i> ; int <i>screen, depth, class</i> ; XVisualInfo * <i>vinfo_return</i> ;	Xlib
int XMaxCmapsOfScreen( <i>screen</i> ) Screen * <i>screen</i> ;	Xlib
XwmenubuttonWidgetClass	Widgets
XwmenumgrWidgetClass	Widgets
XwmenupaneWidgetClass	Widgets
void MenuPopdown( <i>shell_name</i> ) String <i>shell_name</i> ;	Intrinsics
void MenuPopup( <i>shell_name</i> ) String <i>shell_name</i> ;	Intrinsics
XwmenuSepWidgetClass	Widgets
ArgList XtMergeArgLists( <i>arg1, num_args1, arg2, num_args2</i> ) ArgList <i>arg1, arg2</i> ; Cardinal <i>num_args1, num_args2</i> ;	Intrinsics
void XrmMergeDatabases( <i>source_db, target_db</i> ) XrmDatabase <i>source_db, target_db</i> ;	Xlib
int XMinCmapsOfScreen( <i>screen</i> ) Screen * <i>screen</i> ;	Xlib
XwMoveFocus	Widgets
XMoveResizeWindow( <i>display, w, x, y, width, height</i> ) Display * <i>display</i> ; Window <i>w</i> ; int <i>x, y</i> ; unsigned int <i>width, height</i> ;	Xlib
void XtMoveWidget( <i>w, x, y</i> ) Widget <i>w</i> ; Position <i>x, y</i> ;	Intrinsics

Function	Manual
XMoveWindow( <i>display</i> , <i>w</i> , <i>x</i> , <i>y</i> ) Display * <i>display</i> ; Window <i>w</i> ; int <i>x</i> , <i>y</i> ; 	Xlib
Widget XtNameToWidget( <i>reference</i> , <i>names</i> ) Widget <i>reference</i> ; String <i>names</i> ; 	Intrinsics
type *XtNew( <i>type</i> ) type; 	Intrinsics
String XtNewString( <i>str</i> ) String <i>str</i> ; 	Intrinsics
XNextEvent( <i>display</i> , <i>event_return</i> ) Display * <i>display</i> ; XEvent * <i>event_return</i> ; 	Xlib
void XtNextEvent( <i>event_return</i> ); XEvent * <i>event_return</i> ; 	Intrinsics
int XNextRequest( <i>display</i> ) Display * <i>display</i> ; 	Xlib
XNoOp( <i>display</i> ) Display * <i>display</i> ; 	Xlib
Cardinal XtNumber( <i>array</i> ) ArrayVariable <i>array</i> ; 	Intrinsics
Cardinal XtOffset( <i>pointer_type</i> , <i>field_name</i> ) Type <i>pointer_type</i> ; Field <i>field_name</i> ; 	Intrinsics
XOffsetRegion( <i>r</i> , <i>dx</i> , <i>dy</i> ) Region <i>r</i> ; int <i>dx</i> , <i>dy</i> ; 	Xlib
Display *XOpenDisplay( <i>display_name</i> ) char * <i>display_name</i> ; 	Xlib



Function	Manual
Display *XtOpenDisplay( <i>app_context</i> , <i>display_name</i> , <i>application_name</i> , <i>application_class</i> , <i>options</i> , <i>num_options</i> , <i>argc</i> , <i>argv</i> ) XtAppContext <i>app_context</i> ; String <i>display_name</i> , <i>application_name</i> , <i>application_class</i> ; XrmOptionDescRec <i>options</i> [ <i>  </i> ]; Cardinal <i>num_options</i> , * <i>argc</i> ; String <i>argv</i> [ <i>  </i> ];	Intrinsics
void XtOverrideTranslations( <i>w</i> , <i>translations</i> ) Widget <i>w</i> ; XtTranslations <i>translations</i> ;	Intrinsics
Boolean XtOwnSelection( <i>w</i> , <i>selection</i> , <i>time</i> , <i>convertProc</i> , <i>loseSelection</i> , <i>doneProc</i> ) Widget <i>w</i> ; Atom <i>selection</i> ; Time <i>time</i> ; XtConvertSelectionProc <i>convertProc</i> ; XtLoseSelectionProc <i>loseSelection</i> ; XtSelectionDoneProc <i>doneProc</i> ;	Intrinsics
XwPanelWidgetClass	Widgets
Widget XtParent( <i>w</i> ) Widget <i>w</i> ;	Intrinsics
XtAccelerators XtParseAcceleratorTable( <i>source</i> ) String <i>source</i> ;	Intrinsics
Status XParseColor( <i>display</i> , <i>cmap</i> , <i>spec</i> , <i>exact_def_return</i> ) Display * <i>display</i> ; Colormap <i>cmap</i> ; char * <i>spec</i> ; XColor * <i>exact_def_return</i> ;	Xlib

Function	Manual
<pre>void XrmParseCommand(<i>db</i>, <i>table</i>, <i>table_count</i>, <i>name</i>, <i>argc_in_out</i>, <i>argv_in_out</i>)     XrmDatabase *<i>db</i>;     XrmOptionDescList <i>table</i>;     int <i>table_count</i>, *<i>argc_in_out</i>;     char *<i>name</i>, **<i>argv_in_out</i>;</pre>	Xlib
<pre>int XParseGeometry(<i>parsestring</i>, <i>x_return</i>, <i>y_return</i>, <i>width_return</i>, <i>height_return</i>)     char *<i>parsestring</i>;     int *<i>x_return</i>, *<i>y_return</i>;     int *<i>width_return</i>, *<i>height_return</i>;</pre>	Xlib
<pre>XtTranslations XtParseTranslationTable(<i>table</i>)     String <i>table</i>;</pre>	Intrinsics
<pre>XPeekEvent(<i>display</i>, <i>event_return</i>)     Display *<i>display</i>;     XEvent *<i>event_return</i>;</pre>	Xlib
<pre>XtPeekEvent(<i>event_return</i>)     XEvent <i>event_return</i>;</pre>	Intrinsics
<pre>XPeekIfEvent(<i>display</i>, <i>event_return</i>, <i>predicate</i>, <i>arg</i>)     Display *<i>display</i>;     XEvent *<i>event_return</i>;     Bool (*<i>predicate</i>)();     char *<i>arg</i>;</pre>	Xlib
<pre>Boolean XtPending()</pre>	Intrinsics
<pre>int XPending(<i>display</i>)     Display *<i>display</i>;</pre>	Xlib
<pre>char *Xpermalloc(<i>size</i>)     unsigned int <i>size</i>;</pre>	Xlib

Function	Manual
<b>int XHPPixmapToFile</b> ( <i>display, pixmap, color_w, x, y, width, height, plane_mask, format, filename</i> ) Display * <i>display</i> ; Pixmap <i>pixmap</i> ; Window <i>color_w</i> ; int <i>x, y, format</i> ; unsigned int <i>width, height</i> ; long <i>plane_mask</i> ; char <i>filename</i> ;	Xlib
<b>int XPlanesOfScreen</b> ( <i>screen</i> ) Screen * <i>screen</i> ;	Xlib
<b>Bool XPointInRegion</b> ( <i>r, x, y</i> ) Region <i>r</i> ; int <i>x, y</i> ;	Xlib
<b>Region XPolygonRegion</b> ( <i>points, n, fill_rule</i> ) XPoint <i>points</i> [ ]; int <i>n, fill_rule</i> ;	Xlib
<b>void XtPopdown</b> ( <i>popup_shell</i> ) Widget <i>popup_shell</i> ;	Intrinsics
<b>void XtPopup</b> ( <i>popup_shell, grab_kind</i> ) Widget <i>popup_shell</i> ; XtGrabKind <i>grab_kind</i> ;	Intrinsics
<b>XwpopupmgrWidgetClass</b>	Widgets
<b>XwprimitiveWidgetClass</b>	Widgets
<b>XHPPrompt</b> ( <i>display, deviceid, prompt</i> ) Display * <i>display</i> ; XID <i>deviceid</i> ; unsigned int <i>prompt</i> ;	Xlib
<b>int XProtocolRevision</b> ( <i>display</i> ) Display * <i>display</i> );	Xlib



Function	Manual
int XProtocolVersion( <i>display</i> ) Display * <i>display</i> ;	Xlib
XwpulldownWidgetClass	Widgets
XwpushButtonWidgetClass	Widgets
XPutBackEvent( <i>display</i> , <i>event</i> ) Display * <i>display</i> ; XEvent * <i>event</i> ;	Xlib
void XrmPutFileDatabase( <i>database</i> , <i>stored_db</i> ) XrmDatabase <i>database</i> ; char * <i>stored_db</i> ;	Xlib
XPutImage( <i>display</i> , <i>d</i> , <i>gc</i> , <i>image</i> , <i>src_x</i> , <i>src_y</i> , <i>dst_x</i> , <i>dst_y</i> , <i>width</i> , <i>height</i> ) Display * <i>display</i> ; Drawable <i>d</i> ; GC <i>gc</i> ; int <i>src_x</i> , <i>src_y</i> , <i>dst_x</i> , <i>dst_y</i> ; unsigned int <i>width</i> , <i>height</i> ;	Xlib
void XrmPutLineResource( <i>database</i> , <i>line</i> ) XrmDatabase * <i>database</i> ; char * <i>line</i> ;	Xlib
int XPutPixel( <i>ximage</i> , <i>x</i> , <i>y</i> , <i>pixel</i> ) XImage * <i>ximage</i> ; int <i>x</i> , <i>y</i> ; unsigned long <i>pixel</i> ;	Xlib
void XrmPutResource( <i>database</i> , <i>specifier</i> , <i>type</i> , <i>value</i> ) XrmDatabase * <i>database</i> ; char * <i>specifier</i> , <i>type</i> ; XrmValue * <i>value</i> ;	Xlib
void XrmPutStringResource( <i>database</i> , <i>specifier</i> , <i>value</i> ) XrmDatabase * <i>database</i> ; char * <i>specifier</i> , * <i>value</i> ;	Xlib

Function	Manual
<pre> Bool XrmQGetResource(<i>database, quark_name, quark_class,</i> <i>quark_type_return, value_return</i>)     XrmDatabase <i>database</i>;     XrmNameList <i>quark_name</i>;     XrmClassList <i>quark_class</i>;     XrmRepresentation <i>*quark_type_return</i>;     XrmValue <i>*value_return</i>; </pre>	Xlib
<pre> Bool XrmQGetSearchList(<i>database, names, classes, list_return, list_length</i>)     XrmDatabase <i>database</i>;     XrmNameList <i>names</i>;     XrmClassList <i>classes</i>;     XrmSearchList <i>list_return</i>;     int <i>list_length</i>; </pre>	Xlib
<pre> Bool XrmQGetSearchResource(<i>list, name, class, type, type_return,</i> <i>value_return</i>)     XrmSearchList <i>list</i>;     XrmName <i>name</i>;     XrmClass <i>class</i>;     XrmRepresentation <i>*type_return</i>;     XrmValue <i>*value_return</i>; </pre>	Xlib
<pre> int XQLength(<i>display</i>)     Display <i>*display</i>; </pre>	Xlib
<pre> void XrmQPutResource(<i>database, bindings, quarks, type, value</i>)     XrmDatabase <i>*database</i>;     XrmBindingList <i>bindings</i>;     XrmQuarkList <i>quarks</i>;     XrmRepresentation <i>type</i>;     XrmValue <i>*value</i>; </pre>	Xlib
<pre> void XrmQPutStringResource(<i>database, bindings, quarks, value</i>)     XrmDatabase <i>*database</i>;     XrmBindingList <i>bindings</i>;     XrmQuarkList <i>quarks</i>;     char <i>*value</i>; </pre>	Xlib

Function	Manual
char *XrmQuarkToString( <i>quark</i> ) XrmQuark <i>quark</i> ;	Xlib
Status XQueryBestCursor( <i>display</i> , <i>d</i> , <i>width</i> , <i>height</i> , <i>width_return</i> , <i>height_return</i> ) Display * <i>display</i> ; Drawable <i>d</i> ; unsigned int <i>width</i> , <i>height</i> ; unsigned int * <i>width_return</i> , * <i>height_return</i> ;	Xlib
Status XQueryBestSize( <i>display</i> , <i>class</i> , <i>which_screen</i> , <i>width</i> , <i>height</i> , <i>width_return</i> , <i>height_return</i> ) Display * <i>display</i> ; int <i>class</i> ; Drawable <i>which_screen</i> ; unsigned int <i>width</i> , <i>height</i> , * <i>width_return</i> , * <i>height_return</i> ;	Xlib
Status XQueryBestStipple( <i>display</i> , <i>which_screen</i> , <i>width</i> , <i>height</i> , <i>width_return</i> , <i>height_return</i> ) Display * <i>display</i> ; Drawable <i>which_screen</i> ; unsigned int <i>width</i> , <i>height</i> , * <i>width_return</i> , * <i>height_return</i> ;	Xlib
Status XQueryBestTile( <i>display</i> , <i>which_screen</i> , <i>width</i> , <i>height</i> , <i>width_return</i> , <i>height_return</i> ) Display * <i>display</i> ; Drawable <i>which_screen</i> ; unsigned int <i>width</i> , <i>height</i> , * <i>width_return</i> , * <i>height_return</i> ;	Xlib
XQueryColor( <i>display</i> , <i>cmap</i> , <i>def_in_out</i> ) Display * <i>display</i> ; Colormap <i>cmap</i> ; XColor * <i>def_in_out</i> ;	Xlib
XQueryColors( <i>display</i> , <i>cmap</i> , <i>defs_in_out</i> , <i>ncolors</i> ) Display * <i>display</i> ; Colormap <i>cmap</i> ; XColor <i>defs_in_out</i> []; int <i>ncolors</i> ;	Xlib



Function	Manual
Bool XQueryExtension( <i>display</i> , <i>name</i> , <i>major_opcode_return</i> , <i>first_event_return</i> , <i>first_error_return</i> ) Display * <i>display</i> ; char * <i>name</i> ; int * <i>major_opcode_return</i> ; int * <i>first_event_return</i> , * <i>first_error_return</i> ;	Xlib
int XHPQueryImageFile( <i>filename</i> , <i>xwd_header_return</i> ) char <i>filename</i> ; XWDFileHeader <i>xwd_header_return</i> ;	Xlib
XFontStruct *XQueryFont( <i>display</i> , <i>font_ID</i> ) Display * <i>display</i> ; XID <i>font_ID</i> ;	Xlib
XtGeometryResult XtQueryGeometry( <i>w</i> , <i>intended</i> , <i>preferred_return</i> ) Widget <i>w</i> ; XtWidgetGeometry * <i>intended</i> , * <i>preferred_return</i> ;	Intrinsics
XQueryKeymap( <i>display</i> , <i>keys_return</i> ) Display * <i>display</i> ; char <i>keys_return</i> [32];	Xlib
Bool XQueryPointer( <i>display</i> , <i>w</i> , <i>root_return</i> , <i>child_return</i> , <i>root_x_return</i> , <i>root_y_return</i> , <i>win_x_return</i> , <i>win_y_return</i> , <i>mask_return</i> ) Display * <i>display</i> ; Window <i>w</i> , * <i>root_return</i> , * <i>child_return</i> ; int * <i>root_x_return</i> , * <i>root_y_return</i> ; int * <i>win_x_return</i> , * <i>win_y_return</i> ; unsigned int * <i>mask_return</i> ;	Xlib
XQueryTextExtents( <i>display</i> , <i>font_ID</i> , <i>string</i> , <i>nchars</i> , <i>direction_return</i> , <i>font_ascent_return</i> , <i>font_descent_return</i> , <i>overall_return</i> ) Display * <i>display</i> ; XID <i>font_ID</i> ; char * <i>string</i> ; int <i>nchars</i> , * <i>direction_return</i> ; int * <i>font_ascent_return</i> , * <i>font_descent_return</i> ; XCharStruct * <i>overall_return</i> ;	Xlib

Function	Manual
<hr/> XQueryTextExtent16( <i>display, font_ID, string, nchars, direction_return, font_ascent_return, font_descent_return, overall_return</i> ) Display *display; XID *font_ID; XChar2b *string; int nchars, *direction_return; int *font_ascent_return, *font_descent_return; XCharStruct *overall_return; <hr/>	Xlib
Status XQueryTree( <i>display, w, root_return, parent_return, children_return, nchildren_return</i> ) Display *display; Window w, *root_return, *parent_return, **children_return; unsigned int *nchildren_return; <hr/>	Xlib
XRaiseWindow( <i>display, w</i> ) Display *display; Window w; <hr/>	Xlib
int XReadBitmapFile( <i>display, d, filename, width_return, height_return, bitmap_return, x_hot_return, y_hot_return</i> ) Display *display; Drawable d; char *filename; unsigned int *width_return, *height_return; Pixmap *bitmap_return; int *x_hot_return, *y_hot_return; <hr/>	Xlib
void XtRealizeWidget( <i>w</i> ) Widget w; <hr/>	Intrinsics
char *XtRealloc( <i>ptr, num</i> ) char *ptr; Cardinal num; <hr/>	Intrinsics
XRebindKeySym( <i>display, keysym, list, mod_count, string, bytes_string</i> ) Display *display; KeySym keysym, list[ ]; int mod_count, bytes_string; unsigned char *string; <hr/>	Xlib

Function	Manual
<code>XRecolorCursor(<i>display</i>, <i>cursor</i>, <i>foreground_color</i>, <i>background_color</i>)</code> <code>Display *display;</code> <code>Cursor cursor;</code> <code>XColor *foreground_color, *background_color;</code>	Xlib
<code>int XRectInRegion(<i>r</i>, <i>x</i>, <i>y</i>, <i>width</i>, <i>height</i>)</code> <code>Region r;</code> <code>int x, y;</code> <code>unsigned int width, height;</code>	Xlib
<code>XRefreshKeyboardMapping(<i>event_map</i>)</code> <code>XMappingEvent *event_map;</code>	Xlib
<code>XtRegisterCaseConverter(<i>dpy</i>, <i>proc</i>, <i>start</i>, <i>stop</i>)</code> <code>Display *dpy;</code> <code>XtCaseProc proc;</code> <code>KeySym start, stop;</code>	Intrinsics
<code>XwRegisterConverters</code>	Widgets
<code>void XtRemoveAllCallbacks(<i>w</i>, <i>callback_name</i>)</code>	Intrinsics
<code>void XtRemoveCallback(<i>w</i>, <i>callback_name</i>, <i>callback</i>, <i>client_data</i>)</code> <code>Widget w;</code> <code>String callback_name;</code> <code>XtCallbackProc callback;</code> <code>caddr_t client_data;</code>	Intrinsics
<code>void XtRemoveCallbacks(<i>w</i>, <i>callback_name</i>, <i>callback</i>)</code> <code>Widget w;</code> <code>String callback_name;</code> <code>XtCallbackList callbacks;</code>	Intrinsics
<code>void XtRemoveEventHandler(<i>w</i>, <i>event_mask</i>, <i>nonmaskable</i>, <i>proc</i>,  <i>client_data</i>)</code> <code>Widget w;</code> <code>EventMask event_mask;</code> <code>Boolean nonmaskable;</code> <code>XtEventHandler proc;</code> <code>caddr_t client_data;</code>	Intrinsics



Function	Manual
<code>XRemoveFromSaveSet(<i>display</i>, <i>w</i>)</code> Display <i>*display</i> ; Window <i>w</i> ;	Xlib
<code>void XtRemoveGrab(<i>w</i>)</code> Widget <i>w</i> ;	Intrinsics
<code>XRemoveHost(<i>display</i>, <i>host</i>)</code> Display <i>*display</i> ; XHostAddress <i>*host</i> ;	Xlib
<code>XRemoveHosts(<i>display</i>, <i>hosts</i>, <i>num_hosts</i>)</code> Display <i>*display</i> ; XHostAddress <i>*hosts</i> ; int <i>num_hosts</i> ;	Xlib
<code>void XtRemoveInput(<i>id</i>)</code> XtInputID <i>id</i> ;	Intrinsics
<code>void XtRemoveRawEventHandler(<i>w</i>, <i>event_mask</i>, <i>nonmaskable</i>, <i>proc</i>,     <i>client_data</i>)</code> Widget <i>w</i> ; EventMask <i>event_mask</i> ; Boolean <i>nonmaskable</i> ; XtEventHandler <i>proc</i> ; caddr_t <i>client_data</i> ;	Intrinsics
<code>void XtRemoveTimeout(<i>timer</i>)</code> XtIntervalId <i>timer</i> ;	Intrinsics
<code>XReparentWindow(<i>display</i>, <i>w</i>, <i>parent</i>, <i>x</i>, <i>y</i>)</code> Display <i>*display</i> ; Window <i>w</i> , <i>parent</i> ; int <i>x</i> , <i>y</i> ;	Xlib
<code>XResetScreenSaver(<i>display</i>)</code> Display <i>*display</i> ;	Xlib
<code>void XtResizeWidget(<i>w</i>, <i>width</i>, <i>height</i>, <i>border_width</i>)</code> Widget <i>w</i> ; Dimension <i>width</i> , <i>height</i> , <i>border_height</i> ;	Intrinsics

Function	Manual
<b>XResizeWindow</b> ( <i>display, w, width, height</i> ) Display * <i>display</i> ; Window <i>w</i> ; unsigned int <i>width, height</i> ; 	Xlib
<b>XRestackWindows</b> ( <i>display, windows, nwindows</i> ) Display * <i>display</i> ; Window <i>windows[ ]</i> ; int <i>nwindows</i> ; 	Xlib
<b>Keysym XHPRoman8ToKeysym</b> ( <i>r8_char</i> ) char <i>r8_char</i> ; 	Xlib
<b>Window XRootWindow</b> ( <i>display, screen_number</i> ) Display * <i>display</i> ; int <i>screen_number</i> ; 	Xlib
<b>Window XRootWindowOfScreen</b> ( <i>screen</i> ) Screen * <i>screen</i> ; 	Xlib
<b>XRotateBuffers</b> ( <i>display, rotate</i> ) Display * <i>display</i> ; int <i>rotate</i> ; 	Xlib
<b>XRotateWindowProperties</b> ( <i>display, w, properties, num_prop, npositions</i> ) Display * <i>display</i> ; Window <i>w</i> ; Atom <i>properties</i> ; int <i>num_prop, npositions</i> ; 	Xlib
<b>XwrowColWidgetClass</b>	Widgets
<b>XwsashWidgetClass</b>	Widgets
<b>int XSaveContext</b> ( <i>display, w, context, data</i> ) Display * <i>display</i> ; Window <i>w</i> ; XContext <i>context</i> ; caddr_t <i>data</i> ; 	Xlib

Function	Manual
Screen *XtScreen( <i>w</i> ) Widget <i>w</i> ;	Intrinsics
int XScreenCount( <i>display</i> ) Display * <i>display</i> ;	Xlib
Screen *XScreenOfDisplay( <i>display</i> , <i>screen_number</i> ) Display * <i>display</i> ; int <i>screen_number</i> ;	Xlib
XwscrollbarWidgetClass	Widgets
XHPSelectExtensionEvent( <i>display</i> , <i>window</i> , <i>deviceid</i> , <i>mask</i> ) Display * <i>display</i> ; Window <i>window</i> ; XID <i>deviceid</i> ; Mask <i>mask</i> ;	Xlib
XSelectInput( <i>display</i> , <i>w</i> , <i>event_mask</i> ) Display * <i>display</i> ; Window <i>w</i> ; long <i>event_mask</i> ;	Xlib
Status XSendEvent( <i>display</i> , <i>w</i> , <i>propagate</i> , <i>event_mask</i> , <i>event_send</i> ) Display * <i>display</i> ; Window <i>w</i> ; Bool <i>propagate</i> ; long <i>event_mask</i> ; XEvent * <i>event_send</i> ;	Xlib
char *XServerVendor( <i>display</i> ) Display * <i>display</i> ;	Xlib
XSetAccessControl( <i>display</i> , <i>mode</i> ) Display * <i>display</i> ; int <i>mode</i> ;	Xlib
int (*XSetAfterFunction( <i>display</i> , <i>proc</i> ))() Display * <i>display</i> ; int (* <i>proc</i> )();	Xlib



Function	Manual
XSetArcMode( <i>display</i> , <i>gc</i> , <i>arc_mode</i> ) Display * <i>display</i> ; GC <i>gc</i> ; int <i>arc_mode</i> ;	Xlib
XtSetArg( <i>arg</i> , <i>name</i> , <i>value</i> ) Arg <i>arg</i> ; String <i>name</i> ; XtArgVal <i>value</i> ;	Intrinsics
XSetBackground( <i>display</i> , <i>gc</i> , <i>background</i> ) Display * <i>display</i> ; GC <i>gc</i> ; unsigned long <i>background</i> ;	Xlib
XSetClassHint( <i>display</i> , <i>w</i> , <i>class_hints</i> ) Display * <i>display</i> ; Window <i>w</i> ; XClassHint * <i>class_hints</i> ;	Xlib
XSetClipMask( <i>display</i> , <i>gc</i> , <i>pixmap</i> ) Display * <i>display</i> ; GC <i>gc</i> ; Pixmap <i>pixmap</i> ;	Xlib
XSetClipOrigin( <i>display</i> , <i>gc</i> , <i>clip_x_origin</i> , <i>clip_y_origin</i> ) Display * <i>display</i> ; GC <i>gc</i> ; int <i>clip_x_origin</i> , <i>clip_y_origin</i> ;	Xlib
XSetClipRectangles( <i>display</i> , <i>gc</i> , <i>clip_x_origin</i> , <i>clip_y_origin</i> , <i>rectangles</i> , <i>n</i> , <i>ordering</i> ) Display * <i>display</i> ; GC <i>gc</i> ; int <i>clip_x_origin</i> , <i>clip_y_origin</i> , <i>n</i> , <i>ordering</i> ; XRectangle <i>rectangles</i> [ ];	Xlib
XSetCloseDownMode( <i>display</i> , <i>close_mode</i> ) Display * <i>display</i> ; int <i>close_mode</i> ;	Xlib

Function	Manual
XSetCommand( <i>display</i> , <i>w</i> , <i>argv</i> , <i>argc</i> ) Display * <i>display</i> ; Window <i>w</i> ; char ** <i>argv</i> ; int <i>argc</i> ;	Xlib
XSetDashes( <i>display</i> , <i>gc</i> , <i>dash_offset</i> , <i>dash_list</i> , <i>n</i> ) Display * <i>display</i> ; GC <i>gc</i> ; int <i>dash_offset</i> , <i>n</i> ; char <i>dash_list</i> ;	Xlib
XHPSetDeviceFocus( <i>display</i> , <i>deviceid</i> , <i>focus</i> , <i>revert_to</i> , <i>time</i> ) Display * <i>display</i> ; XID <i>deviceid</i> ; Window <i>focus</i> ; int <i>revert_to</i> ; Time <i>time</i> ;	Xlib
XHPSetDeviceModifierMapping( <i>display</i> , <i>deviceid</i> , <i>modmap</i> ) Display * <i>display</i> ; XID <i>deviceid</i> ; XModifierKeymap * <i>modmap</i> ;	Xlib
XSetErrorHandler( <i>handler</i> ) int (* <i>handler</i> ) (Display *, XErrorEvent *);	Xlib
PFI XHPSetErrorHandler( <i>display</i> , <i>routine</i> ) Display * <i>display</i> ; int (* <i>routine</i> ) ();	Xlib
void XtSetErrorMsgHandler( <i>msg_handler</i> ) XtErrorMsgHandler <i>msg_handler</i> ;	Intrinsics
XSetFillRule( <i>display</i> , <i>gc</i> , <i>fill_rule</i> ) Display * <i>display</i> ; GC <i>gc</i> ; int <i>fill_rule</i> ;	Xlib
XSetFillStyle( <i>display</i> , <i>gc</i> , <i>fill_style</i> ) Display * <i>display</i> ; GC <i>gc</i> ; int <i>fill_style</i> ;	Xlib

Function	Manual
<b>XSetFont</b> ( <i>display, gc, font</i> ) Display * <i>display</i> ; GC <i>gc</i> ; Font <i>font</i> ; 	Xlib
<b>XSetFontPath</b> ( <i>display, directories, ndirs</i> ) Display * <i>display</i> ; char ** <i>directories</i> ; int <i>ndirs</i> ; 	Xlib
<b>XSetForeground</b> ( <i>display, gc, foreground</i> ) Display * <i>display</i> ; GC <i>gc</i> ; unsigned long <i>foreground</i> ; 	Xlib
<b>XSetFunction</b> ( <i>display, gc, function</i> ) Display * <i>display</i> ; GC <i>gc</i> ; int <i>function</i> ; 	Xlib
<b>XSetGraphicsExposures</b> ( <i>display, gc, graphics_exposures</i> ) Display * <i>display</i> ; GC <i>gc</i> ; Bool <i>graphics_exposures</i> ; 	Xlib
<b>XSetIconName</b> ( <i>display, w, icon_name</i> ) Display * <i>display</i> ; Window <i>w</i> ; char * <i>icon_name</i> ; 	Xlib
<b>XSetIconSizes</b> ( <i>display, w, size_list, count</i> ) Display * <i>display</i> ; Window <i>w</i> ; XIconSize * <i>size_list</i> ; int <i>count</i> ; 	Xlib



Function	Manual
<pre>int XHPSetInputDevice(<i>display, deviceid, mode</i>)     Display *<i>display</i>;     XID <i>deviceid</i>;     int <i>mode</i>;</pre>	Xlib
<pre>XSetInputFocus(<i>display, focus, revert_to, time</i>)     Display *<i>display</i>;     Window <i>focus</i>;     int <i>revert_to</i>;     Time <i>time</i>;</pre>	Xlib
<pre>XSetIOErrorHandler(<i>handlers</i>)     int (*<i>handler</i>)(Display *);</pre>	Xlib
<pre>XtSetKeyboardFocus(<i>w, descendent</i>)     Widget <i>w, descendent</i>;</pre>	Intrinsics
<pre>void XtSetKeyTranslator(<i>dpy, proc</i>)     display *<i>dpy</i>;     XtKeyProc <i>proc</i>;</pre>	Intrinsics
<pre>XSetLineAttributes(<i>display, gc, line_width, line_style, cap_style, join_style</i>)     Display *<i>display</i>;     GC <i>gc</i>;     unsigned int <i>line_width</i>;     int <i>line_style, cap_style, join_style</i>;</pre>	Xlib
<pre>void XtSetMappedWhenManaged(<i>w, map_when_managed</i>)     Widget <i>w</i>;     Boolean <i>map_when_managed</i>;</pre>	Intrinsics
<pre>XSetModifierMapping(<i>display, modmap</i>)     Display *<i>display</i>;     XModifierKeymap *<i>modmap</i>;</pre>	Xlib
<pre>void XSetNormalHints(<i>display, w, hints</i>)     Display *<i>display</i>;     Window <i>w</i>;     XSizeHints *<i>hints</i>;</pre>	Xlib

Function	Manual
XSetPlaneMask( <i>display, gc, plane_mask</i> ) Display * <i>display</i> ; GC <i>gc</i> ; unsigned long <i>plane_mask</i> ;	Xlib
int XSetPointerMapping( <i>display, map, nmap</i> ) Display * <i>display</i> ; unsigned char <i>map</i> [ ]; int <i>nmap</i> ;	Xlib
XSetRegion( <i>display, gc, r</i> ) Display * <i>display</i> ; GC <i>gc</i> ; Region <i>r</i> ;	Xlib
XSetScreenSaver( <i>display, timeout, interval, prefer_blanking, allow_exposures</i> ) Display * <i>display</i> ; int <i>timeout, interval, prefer_blanking, allow_exposures</i> ;	Xlib
XSetSelectionOwner( <i>display, selection, owner, time</i> ) Display * <i>display</i> ; Atom <i>selection</i> ; Window <i>owner</i> ; Time <i>time</i> ;	Xlib
void XtSetSelectionTimeout( <i>timeout</i> ); unsigned long <i>timeout</i> ;	Intrinsics
void XtSetSensitive( <i>w, sensitive</i> ) Widget <i>w</i> ; Boolean <i>sensitive</i> ;	Intrinsics
XSetSizeHints( <i>display, w, hints, property</i> ) Display * <i>display</i> ; Window <i>w</i> ; XSizeHints * <i>hints</i> ; Atom <i>property</i> ;	Xlib

Function	Manual
<pre>void XSetStandardColormap(<i>display, w, cmap, property</i>)     Display *<i>display</i>;     Window <i>w</i>;     XStandardColormap *<i>cmap</i>;     Atom <i>property</i>;</pre>	Xlib
<pre>XSetStandardProperties(<i>display, w, window_name, icon_name, icon_pixmap, argv, argc, hints</i>)     Display *<i>display</i>;     Window <i>w</i>;     char *<i>window_name</i>, *<i>icon_name</i>, **<i>argv</i>;     Pixmap <i>icon_pixmap</i>;     int <i>argc</i>;     XSizeHints *<i>hints</i>;</pre>	Xlib
<pre>XSetState(<i>display, gc, foreground, background, function, plane_mask</i>)     Display *<i>display</i>;     GC <i>gc</i>;     unsigned long <i>foreground, background, plane_mask</i>;     int <i>function</i>;</pre>	Xlib
<pre>XSetStipple(<i>display, gc, stipple</i>)     Display *<i>display</i>;     GC <i>gc</i>;     Pixmap <i>stipple</i>;</pre>	Xlib
<pre>void XtSetSubvalues(<i>base, resources, num_resources, args, num_args</i>)     caddr_t <i>base</i>;     XtResourceList <i>resources</i>;     Cardinal <i>num_resources, num_args</i>;     ArgList <i>args</i>;</pre>	Intrinsics
<pre>XSetSubwindowMode(<i>display, gc, subwindow_mode</i>)     Display *<i>display</i>;     GC <i>gc</i>;     int <i>subwindow_mode</i>;</pre>	Xlib



Function	Manual
XSetTile( <i>display</i> , <i>gc</i> , <i>tile</i> ) Display * <i>display</i> ; GC <i>gc</i> ; Pixmap <i>tile</i> ;	Xlib
XSetTransientForHint( <i>display</i> , <i>w</i> , <i>prop_window</i> ) Display * <i>display</i> ; Window <i>w</i> ; Window <i>prop_window</i> ;	Xlib
XSetTSOrigin( <i>display</i> , <i>gc</i> , <i>ts_x_origin</i> , <i>ts_y_origin</i> ) Display * <i>display</i> ; GC <i>gc</i> ; int <i>ts_x_origin</i> , <i>ts_y_origin</i> ;	Xlib
void XtSetValues( <i>w</i> , <i>args</i> , <i>num_args</i> ) Widget <i>w</i> ; ArgList <i>args</i> ; Cardinal <i>num_args</i> ;	Intrinsics
void XtSetWarningMsgHandler( <i>msg_handler</i> ) XtErrorMsgHandler <i>msg_handler</i> ;	Intrinsics
XSetWindowBackground( <i>display</i> , <i>w</i> , <i>background_pixel</i> ) Display * <i>display</i> ; Window <i>w</i> ; unsigned long <i>background_pixel</i> ;	Xlib
XSetWindowBackgroundPixmap( <i>display</i> , <i>w</i> , <i>background_pixmap</i> ) Display * <i>display</i> ; Window <i>w</i> ; Pixmap <i>background_pixmap</i> ;	Xlib
XSetWindowBorder( <i>display</i> , <i>w</i> , <i>border_pixel</i> ) Display * <i>display</i> ; Window <i>w</i> ; unsigned long <i>border_pixel</i> ;	Xlib

Function	Manual
XSetWindowBorderPixmap( <i>display</i> , <i>w</i> , <i>border_pixmap</i> ) Display * <i>display</i> ; Window <i>w</i> ; Pixmap <i>border_pixmap</i> ;	Xlib
XSetWindowBorderWidth( <i>display</i> , <i>w</i> , <i>width</i> ) Display * <i>display</i> ; Window <i>w</i> ; unsigned int <i>width</i> ;	Xlib
XSetWindowColormap( <i>display</i> , <i>w</i> , <i>cmap</i> ) Display * <i>display</i> ; Window <i>w</i> ; Colormap <i>cmap</i> ;	Xlib
XSetWMHints( <i>display</i> , <i>w</i> , <i>wmhints</i> ) Display * <i>display</i> ; Window <i>w</i> ; XWMHints * <i>wmhints</i> ;	Xlib
XSetZoomHints( <i>display</i> , <i>w</i> , <i>zhints</i> ) Display * <i>display</i> ; Window <i>w</i> ; XSizeHints * <i>zhints</i> ;	Xlib
XShrinkRegion( <i>r</i> , <i>dx</i> , <i>dy</i> ) Region <i>r</i> ; int <i>dy</i> , <i>dy</i> ;	Xlib
XwstaticcrasterWidgetClass	Widgets
XwstatictextWidgetClass	Widgets
XStoreBuffer( <i>display</i> , <i>bytes</i> , <i>nbytes</i> , <i>buffer</i> ) Display * <i>display</i> ; char * <i>bytes</i> ; int <i>nbytes</i> , <i>buffer</i> ;	Xlib

Function	Manual
XStoreBytes( <i>display, bytes, nbytes</i> ) Display * <i>display</i> ; char * <i>bytes</i> ; int <i>nbytes</i> ; 	Xlib
XStoreColor( <i>display, cmap, color</i> ) Display * <i>display</i> ; Colormap <i>cmap</i> ; XColor <i>color</i> ; 	Xlib
XStoreColors( <i>display, cmap, color, ncolors</i> ) Display * <i>display</i> ; Colormap <i>cmap</i> ; XColor <i>color</i> []; int <i>ncolors</i> ; 	Xlib
XStoreName( <i>display, w, window_name</i> ) Display * <i>display</i> ; Window <i>w</i> ; char * <i>window_name</i> ; 	Xlib
XStoreNamedColor( <i>display, cmap, color, pixel, flags</i> ) Display * <i>display</i> ; Colormap <i>cmap</i> ; char * <i>color</i> ; unsigned long <i>pixel</i> ; int <i>flags</i> ; 	Xlib
XrmStringToBindingQuarkList( <i>string, bindings_return, quarks_return</i> ) char * <i>string</i> ; XrmBindingList <i>bindings_return</i> ; XrmQuarkList <i>quarks_return</i> ; 	Xlib
KeySym XStringToKeysym( <i>string</i> ) char * <i>string</i> ; 	Xlib
XrmQuark XrmStringToQuark( <i>string</i> ) char * <i>string</i> ; 	Xlib



Function	Manual
void XrmStringToQuarkList( <i>string, quarks_return</i> ) char * <i>string</i> ; XrmQuarkList <i>quarks_return</i> ;	Xlib
XImage *XSubImage( <i>ximage, x, y, subimage_width, subimage_height</i> ) XImage * <i>ximage</i> ; int <i>x, y</i> ; unsigned int <i>subimage_width, subimage_height</i> ;	Xlib
XSubtractRegion( <i>sra, srb, dr_return</i> ) Region <i>sra, srb, dr_return</i> ;	Xlib
WidgetClass XtSuperclass( <i>w</i> ) Widget <i>w</i> ;	Intrinsics
XtswindowWidgetClass	Widgets
XSync( <i>display, discard</i> ) Display * <i>display</i> ; Bool <i>discard</i> ;	Xlib
int (*XSynchronize( <i>display, onoff</i> ))() Display * <i>display</i> ; int <i>onoff</i> ;	Xlib
XttexteditWidgetClass	Widgets
XTextExtents( <i>font_struct, string, nchars, direction_return,</i> <i>font_ascent_return, font_descent_return, overall_return</i> ) XFontStruct * <i>font_struct</i> ; char * <i>string</i> ; int <i>nchars, *directional_return</i> ; int * <i>font_ascent_return, *font_descent_return</i> ; XCharStruct * <i>overall_return</i> ;	Xlib
XTextExtents16( <i>font_struct, string, nchars, direction_return,</i> <i>font_ascent_return, font_descent_return, overall_return</i> ) XFontStruct * <i>font_struct</i> ; XChar2b * <i>string</i> ; int <i>nchars, *direction_return</i> ; int * <i>font_ascent_return, *font_descent_return</i> ; XCharStruct * <i>overall_return</i> ;	Xlib

Function	Manual
<pre>int XTextWidth(<i>font_struct</i>, <i>string</i>, <i>count</i>)     XFontStruct *<i>font_struct</i>;     char *<i>string</i>;     int <i>count</i>;</pre>	Xlib
<pre>int XTextWidth16(<i>font_struct</i>, <i>string</i>, <i>count</i>)     XFontStruct *<i>font_struct</i>;     XChar2b *<i>string</i>;     int <i>count</i>;</pre>	Xlib
XwttitlebarWidgetClass	Widgets
XwtoggleWidgetClass	Widgets
<pre>Bool XTranslateCoordinates(<i>display</i>, <i>src_w</i>, <i>dest_w</i>, <i>src_x</i>, <i>src_y</i>,     <i>dest_x_return</i>, <i>dest_y_return</i>, <i>child_return</i>)     Display *<i>display</i>;     Window <i>src_w</i>, <i>dest_w</i>, *<i>child_return</i>;     int <i>src_x</i>, <i>src_y</i>, *<i>dest_x_return</i>, *<i>dest_y_return</i>;</pre>	Xlib
<pre>void XtTranslateCoords(<i>w</i>, <i>x</i>, <i>y</i>, <i>rootx_return</i>, <i>rooty_return</i>)     Widget <i>w</i>;     Position <i>x</i>, <i>y</i>, *<i>rootx_return</i>, *<i>rooty_return</i>;</pre>	Intrinsics
<pre>XUndefineCursor(<i>display</i>, <i>w</i>)     Display *<i>display</i>;     Window <i>w</i>;</pre>	Xlib
<pre>XUngrabButton(<i>display</i>, <i>button_ungrab</i>, <i>modifiers</i>, <i>ungrab_window</i>)     Display *<i>display</i>;     unsigned int <i>button_ungrab</i>, <i>modifiers</i>;     Window <i>ungrab_window</i>;</pre>	Xlib
<pre>*XHPUngrabDevice(<i>display</i>, <i>deviceid</i>, <i>time</i>)     Display *<i>display</i>;     XID <i>deviceid</i>;     Time <i>time</i>;</pre>	Xlib

Function	Manual
UngrabDevicebutton( <i>display, deviceid, button, modifiers, ungrab_window</i> ) Display * <i>display</i> ; XID <i>deviceid</i> ; unsigned int <i>button, modifiers</i> ; Window <i>ungrab_window</i> ;	Xlib
XHPUngrabDeviceKey( <i>display, deviceid, keycode, modifiers, ungrab_window</i> ) Display * <i>display</i> ; XID <i>deviceid</i> ; unsigned int <i>keycode, modifiers</i> ; Window <i>ungrab_window</i> ;	Xlib
XUngrabKey( <i>display, keycode, modifiers, ungrab_window</i> ) Display * <i>display</i> ; int <i>keycode</i> ; unsigned int <i>modifiers</i> ; Window <i>ungrab_window</i> ;	Xlib
XUngrabKeyboard( <i>display, time</i> ) Display * <i>display</i> ; Time <i>time</i> ;	Xlib
XUngrabPointer( <i>display, time</i> ) Display * <i>display</i> ; Time <i>time</i> ;	Xlib
XUngrabServer( <i>display</i> ) Display * <i>display</i> ;	Xlib
XUninstallColormap( <i>display, cmap</i> ) Display * <i>display</i> ; Colormap <i>cmap</i> ;	Xlib
void XtUninstallTranslations( <i>w</i> ) Widgets <i>w</i> ;	Intrinsics
XUnionRectWithRegion( <i>rectangle, src_region, dest_region_return</i> ) Rectangle * <i>rectangle</i> ; Region <i>src_region, dest_region_return</i> ;	Xlib



Function	Manual
<code>XUnionRegion(<i>sra</i>, <i>srb</i>, <i>dr_return</i>)</code> Region <i>sra</i> , <i>srb</i> , <i>dr_return</i> ;	Xlib
<code>XrmQuark XrmUniqueQuark()</code>	Xlib
<code>XUnloadFont(<i>display</i>, <i>font</i>)</code> Display <i>*display</i> ; Font <i>font</i> ;	Xlib
<code>UnlockDisplay(<i>display</i>)</code> Display <i>*display</i> ;	Xlib
<code>void XtUnmanageChild(<i>child</i>)</code> Widget <i>child</i> ;	Intrinsics
<code>void XtUnmanageChildren(<i>children</i>, <i>num_children</i>)</code> WidgetList <i>children</i> ; Cardinal <i>num_children</i> ;	Intrinsics
<code>XUnmapSubwindows(<i>display</i>, <i>w</i>)</code> Display <i>*display</i> ; Window <i>w</i> ;	Xlib
<code>XtUnmapWidget(<i>w</i>)</code> Widget <i>w</i> ;	Intrinsics
<code>XUnmapWindow(<i>display</i>, <i>w</i>)</code> Display <i>*display</i> ; Window <i>w</i> ;	Xlib
<code>void XtUnrealizeWidget(<i>w</i>)</code> Widget <i>w</i> ;	Intrinsics
<code>XwvaluatorWidgetClass</code>	Widgets
<code>int XVendorRelease(<i>display</i>)</code> Display <i>*display</i> ;	Xlib
<code>XwvPanedWidgetClass</code>	Widgets
<code>void XtWarning(<i>message</i>)</code> String <i>message</i> ;	Intrinsics

Function	Manual
void XtWarningMsg( <i>name, type, class, defaultp, params, num_params</i> ) String <i>name, type, class, defaultp, *params</i> ; Cardinal <i>num_params</i> ;	Intrinsics
XWarpPointer( <i>display, src_w, dest_w, src_x, src_y, src_width, src_height, dest_x, dest_y</i> ) Display <i>*display</i> ; Window <i>src_w, dest_w</i> ; int <i>src_x, src_y, dest_x, dest_y</i> ; unsigned int <i>src_width, src_height</i> ;	Xlib
unsigned long XWhitePixel( <i>display, screen_number</i> ) Display <i>*display</i> ; int <i>screen_number</i> ;	Xlib
unsigned long XWhitePixelOfScreen( <i>screen</i> ) Screen <i>*screen</i> ;	Xlib
int XWidthMMOfScreen( <i>screen</i> ) Screen <i>*screen</i> ;	Xlib
int XWidthOfScreen( <i>screen</i> ) Screen <i>*screen</i> ;	Xlib
XtAppContext XtWidgetToApplicationContext( <i>w</i> ) Widget <i>w</i> ;	Intrinsics
Window XtWindow( <i>w</i> ) Widget <i>w</i> ;	Intrinsics
XWindowEvent( <i>display, w, event_mask, event_return</i> ) Display <i>*display</i> ; Window <i>w</i> ; long <i>event_mask</i> ; XEvent <i>*event_return</i> ;	Xlib
int XHPWindowToFile( <i>display, w, x, y, width, height, format, filename</i> ) Display <i>*display</i> ; Window <i>w</i> ; int <i>x, y, format</i> ; unsigned int <i>width, height</i> ; char <i>filename</i> ;	Xlib

Function	Manual
Widget XtWindowToWidget( <i>display, window</i> ) Display * <i>display</i> ; Window <i>window</i> ;	Intrinsics
XworkSpaceWidgetClass	Widgets
int XWriteBitmapFile( <i>display, filename, bitmap, width, height, x_hot, y_hot</i> ) Display * <i>display</i> ; Drawable <i>d</i> ; char * <i>filename</i> ; Pixmap <i>bitmap</i> ; unsigned int <i>width, height</i> ; int <i>x_hot, y_hot</i> ;	Xlib
XXorRegion( <i>sra, srb, dr_return</i> ) Region <i>sra, srb, dr_return</i> ;	Xlib



## Functions Sorted by Description

---

The items in this chapter are sorted in alphabetical order by the verb describing the function.

Description	Function	Manual
Achieve effects of X10 XDraw, XDrawDashed, XDrawPatterned	Status XDraw( <i>display, d, gc, vlist, vcount</i> )	Xlib
Achieve effects of X10 XDrawTiled and XDrawFilled	Status XDrawFilled( <i>display, d, gc, vlist, vcount</i> )	Xlib
Activate the screen saver	XActivateScreenSaver( <i>display</i> )	Xlib
Add callback procedure	void XtAddCallback( <i>w, callback_name, callback, client_data</i> )	Intrinsics
Add child to parent's array	void InsertChild( <i>w</i> )	Intrinsics
Add extension data structure	XAddToExtensionList( <i>structure, ext_data</i> )	Xlib
Add list of callback procedures to a callback list	void XtAddCallbacks( <i>w, callback_name, callbacks</i> )	Intrinsics
Add list of children to their parent's managed set	void XtManageChildren( <i>children, num_children</i> )	Intrinsics
Add multiple hosts	XAddHosts( <i>display, hosts, num_hosts</i> )	Xlib
Add new entry to an XModifierKeymap structure	XModifierKeymap *XInsertModifiermapEntry( <i>modmap, keycode_entry, modifier</i> )	Xlib
Add or remove a window from the client's save-set	XChangeSaveSet( <i>display, w, change_mode</i> )	Xlib

<b>Description</b>	<b>Function</b>	<b>Manual</b>
Add resource that is specified as a string to the database	<code>void XrmPutStringResource(<i>database</i>, <i>specifier</i>, <i>value</i>)</code>	Xlib
Add single child to parent's managed set	<code>void XtManageChild(<i>child</i>)</code>	Intrinsics
Add single host	<code>XAddHost(<i>display</i>)</code>	Xlib
Add single resource entry that is specified as a string that contains both a name and a value to the database	<code>void XrmPutLineResource(<i>database</i>, <i>line</i>)</code>	Xlib
Add string resource using quarks as a specification to the database	<code>void XrmQPutStringResource(<i>database</i>, <i>bindings</i>, <i>quarks</i>, <i>value</i>)</code>	Xlib
Add window to the client's save-set	<code>XAddToSaveSet(<i>display</i>, <i>w</i>)</code>	Xlib
Allocate and initialize an array	<code>char *XtCalloc(<i>num</i>, <i>size</i>)</code>	Intrinsics
Allocate memory that won't be given back	<code>char *Xpermalloc(<i>size</i>)</code>	Xlib
Allocate new quark	<code>XrmQuark XrmUniqueQuark()</code>	Xlib
Allocate read-only color cell	<code>Status XAllocColor(<i>display</i>, <i>cmap</i>, <i>screen_in_out</i>)</code>	Xlib
Allocate read-only color cell by name and return the closest color supported by the hardware	<code>Status XAllocNamedColor(<i>display</i>, <i>cmap</i>, <i>color_name</i>, <i>screen_def_return</i>, <i>exact_def_return</i>)</code>	Xlib
Allocate read/write color cells and color plane combinations for a Pseudocolor model	<code>Status XAllocColorCells(<i>display</i>, <i>cmap</i>, <i>contig</i>, <i>plane_masks_return</i>, <i>nplanes</i>, <i>pixels_return</i>, <i>ncolors</i>)</code>	Xlib
Allocate read/write color resources for a DirectColor model	<code>Status XAllocColorPlanes(<i>display</i>, <i>cmap</i>, <i>contig</i>, <i>pixels_return</i>, <i>ncolors</i>, <i>nreds</i>, <i>ngreens</i>, <i>nblues</i>, <i>rmask_return</i>, <i>gmask_return</i>, <i>bmask_return</i>)</code>	Xlib

## 2-2 Functions Sorted by Description

Description	Function	Manual
Allocate storage	<code>char *XtMalloc(<i>size</i>)</code>	Intrinsics
Allocate storage for a new instance of a data type	<code>type *XtNew(<i>type</i>)</code>	Intrinsics
Allocate sufficient memory for an XImage structure	<code>XImage *XCreateImage(<i>display</i>, <i>visual</i>, <i>depth</i>, <i>format</i>, <i>offset</i>, <i>data</i>, <i>width</i>, <i>height</i>, <i>bitmap_pat</i>, <i>bytes_per_line</i>)</code>	Xlib
Allow further events to be processed when the device has been frozen	<code>XAllowEvents(<i>display</i>, <i>event_mode</i>, <i>time</i>)</code>	Xlib
Calculate the difference between the union and the intersection of two regions	<code>XXorRegion(<i>sra</i>, <i>srb</i>, <i>dr_return</i>)</code>	Xlib
Call a high level error handler	<code>void XtErrorMsg(<i>name</i>, <i>type</i>, <i>class</i>, <i>defaultp</i>, <i>params</i>, <i>num_params</i>)</code>	Intrinsics
Call a high level warning handler	<code>void XtWarningMsg(<i>name</i>, <i>type</i>, <i>class</i>, <i>defaultp</i>, <i>params</i>, <i>num_params</i>)</code>	Intrinsics
Call an installed fatal error procedure	<code>void XtError(<i>message</i>)</code>	Intrinsics
Call an installed non-fatal error procedure	<code>void XtWarning(<i>message</i>)</code>	Intrinsics
Change access control	<code>XSetAccessControl(<i>display</i>, <i>mode</i>)</code>	Xlib
Change active pointer grab	<code>XChangeActivePointerGrab(<i>display</i>, <i>event_mask</i>, <i>cursor</i>, <i>time</i>)</code>	Xlib
Change and repaint a window's border tile	<code>XSetWindowBorderPixmap(<i>display</i>, <i>w</i>, <i>border_pixmap</i>)</code>	Xlib
Change and repaint a window's border to a given pixel	<code>XSetWindowBorder(<i>display</i>, <i>w</i>, <i>border_pixel</i>)</code>	Xlib
Change border of window	<code>XSetWindowBorderWidth(<i>display</i>, <i>w</i>, <i>width</i>)</code>	Xlib



<b>Description</b>	<b>Function</b>	<b>Manual</b>
Change close down mode of the client	<code>XSetCloseDownMode(<i>display</i>, <i>close_mode</i>)</code>	Xlib
Change color of a cursor	<code>XRecolorCursor(<i>display</i>, <i>cursor</i>, <i>foreground_color</i>, <i>background_color</i>)</code>	Xlib
Change components in a graphics context	<code>XChangeGC(<i>display</i>, <i>gc</i>, <i>valuemask_change</i>, <i>values</i>)</code>	Xlib
Change control attributes of an extension input device	<code>XHPChangeDeviceControl(<i>display</i>, <i>deviceid</i>, <i>value_mask</i>, <i>values</i>)</code>	Xlib
Change control of a keyboard	<code>XChangeKeyboardControl(<i>display</i>, <i>value_mask</i>, <i>values</i>)</code>	Xlib
Change key mappiong of an extension input device	<code>XHPChangeDeviceKeyMapping(<i>display</i>, <i>deviceid</i>, <i>first_keycode</i>, <i>keysyms_per_keycode</i>, <i>keysyms</i>, <i>num_codes</i>)</code>	Xlib
Change keyboard mapping	<code>XChangeKeyboardMapping(<i>display</i>, <i>first_keycode</i>, <i>keysyms_per_keycode</i>, <i>keysyms</i>, <i>num_codes</i>)</code>	Xlib
Change map_when_managed field	<code>void XtSetMappedWhenManaged(<i>w</i>, <i>map_when_managed</i>)</code>	Intrinsics
Change modifier mapping of an extension input device	<code>XHPSetDeviceModifierMapping(<i>display</i>, <i>deviceid</i>, <i>modmap</i>)</code>	Xlib
Change one or more window attributes	<code>XChangeWindowAttributes(<i>display</i>, <i>w</i>, <i>valuemask</i>, <i>attributes</i>)</code>	Xlib
Change property of a window	<code>XChangeProperty(<i>display</i>, <i>w</i>, <i>property</i>, <i>type</i>, <i>format</i>, <i>mode</i>, <i>data</i>, <i>nelements</i>)</code>	Xlib
Change size of an allocated block of storage	<code>char *XtRealloc(<i>ptr</i>, <i>num</i>)</code>	Intrinsics
Change window's parent within a single screen	<code>XReparentWindow(<i>display</i>, <i>w</i>, <i>parent</i>, <i>x</i>, <i>y</i>)</code>	Xlib

<b>Description</b>	<b>Function</b>	<b>Manual</b>
Change window's size and location	<code>XMoveResizeWindow(<i>display</i>, <i>w</i>, <i>x</i>, <i>y</i>, <i>width</i>, <i>height</i>)</code>	Xlib
Change window's size without changing upper left coordinate	<code>XResizeWindow(<i>display</i>, <i>w</i>, <i>width</i>, <i>height</i>)</code>	Xlib
Check current sensitivity state of a widget	<code>Boolean XtIsSensitive(<i>w</i>)</code>	Intrinsics
Check event queue for a matching event and, if found, remove the event from the queue	<code>XIfEvent(<i>display</i>, <i>event_return</i>, <i>predicate</i>, <i>arg</i>)</code>	Xlib
Check event queue for a matching event but not remove the event from the queue	<code>XPeekIfEvent(<i>display</i>, <i>event_return</i>, <i>predicate</i>, <i>arg</i>)</code>	Xlib
Check event queue for a matching event without blocking	<code>Bool XCheckIfEvent(<i>display</i>, <i>event_return</i>, <i>predicate</i>, <i>arg</i>)</code>	Xlib
Check if KeySym is a cursor key	<code>IsCursorKey(<i>keysym</i>)</code>	Xlib
Check if KeySym is a function key	<code>IsFunctionKey(<i>keysym</i>)</code>	Xlib
Check if KeySym is a keypad key	<code>IsKeypadKey(<i>keysym</i>)</code>	Xlib
Check if KeySym is a miscellaneous function key	<code>IsMiscFunctionKey(<i>keysym</i>)</code>	Xlib
Check if KeySym is a modifier key	<code>IsModifierKey(<i>keysym</i>)</code>	Xlib
Check if point is in a region	<code>Bool XPointInRegion(<i>r</i>, <i>x</i>, <i>y</i>)</code>	Xlib
Check if rectangle is in a region	<code>int XRectInRegion(<i>r</i>, <i>x</i>, <i>y</i>, <i>width</i>, <i>height</i>)</code>	Xlib
Check if region is empty	<code>Bool XEmptyRegion(<i>r</i>)</code>	Xlib

<b>Description</b>	<b>Function</b>	<b>Manual</b>
Check if there are any events on the input queue	Boolean <code>XtAppPending(<i>app_context</i>)</code>	Intrinsics
Check if there are events pending on the queue	Boolean <code>XtPending()</code>	Intrinsics
Check if two bytes are defined as a 16-bit character for a specified font	Bool <code>XHPis16bitCharacter(<i>font</i>, <i>byte1</i>, <i>byte2</i>)</code>	Xlib
Check if two regions have the same offset, size, and shape	Bool <code>XEqualRegion(<i>r1</i>, <i>r2</i>)</code>	Xlib
Check if widget has been realized	Boolean <code>XtIsRealized(<i>w</i>)</code>	Intrinsics
Check if widget owns a selection	Boolean <code>XtOwnSelection(<i>w</i>, <i>selection</i>, <i>time</i>, <i>convertProc</i>, <i>loseSelection</i>, <i>doneProc</i>)</code>	Intrinsics
Check number of events in the event queue	int <code>XEventQueued(<i>display</i>, <i>mode</i>)</code>	Xlib
Check status of callback list	<code>XtCallbackStatus</code> <code>XtHasCallbacks(<i>w</i>, <i>callback_name</i>)</code>	Intrinsics
Check subclass of a widget	<code>XtIsSubclass(<i>w</i>, <i>widget_class</i>)</code>	Intrinsics
Check subclass of widget and generate message	void <code>XtCheckSubclass(<i>w</i>, <i>widget_class</i>, <i>message</i>)</code>	Intrinsics
Circulate a subwindow up or down	<code>XCirculateSubwindows(<i>display</i>, <i>w</i>, <i>direction</i>)</code>	Xlib
Clear entire area within a window	<code>XCclearWindow(<i>display</i>, <i>w</i>)</code>	Xlib
Clear rectangular area of a window	<code>XCclearArea(<i>display</i>, <i>w</i>, <i>x</i>, <i>y</i>, <i>width</i>, <i>height</i>, <i>exposures</i>)</code>	Xlib
Clear timeout value	void <code>XtRemoveTimeOut(<i>timer</i>)</code>	Intrinsics
Close display	void <code>XtCloseDisplay(<i>display</i>)</code>	Intrinsics
Close display	<code>XCcloseDisplay(<i>display</i>)</code>	Xlib



<b>Description</b>	<b>Function</b>	<b>Manual</b>
Combine image in memory with rectangle of a drawable on the display	<code>XPutImage(<i>display, d, gc, image, src_x, src_y, dst_x, dst_y, width, height</i>)</code>	Xlib
Compile a translation table	<code>XtTranslations</code> <code>XtParseTranslationTable(<i>table</i>)</code>	Intrinsics
Compute the intersection of two regions	<code>XIntersectRegion(<i>sra, srb, dr_return</i>)</code>	Xlib
Compute the union of two regions	<code>XUnionRegion(<i>sra, srb, dr_return</i>)</code>	Xlib
Configure a window's size, location, stacking, or border	<code>XConfigureWindow(<i>display, w, value_mask, values</i>)</code>	Xlib
Control size of panes widget	<code>XwsashWidgetClass</code>	Widgets
Control the interactive feel of the pointer	<code>XChangePointerControl(<i>display, do_accel, do_threshold, accel_numerator, accel_denominator, threshold</i>)</code>	Xlib
Convert between HP Roman 8 characters and X KeySyms	<code>Keysym XHPRoman8ToKeysym(<i>r8_char</i>)</code>	Xlib
Convert between X KeySyms and HP Roman 8 characters	<code>int XHPKeysymToRoman8(<i>keysym, r8_return</i>)</code>	Xlib
Convert KeyCode to a defined KeySym	<code>KeySym XKeycodeToKeysym(<i>display, keycode, index</i>)</code>	Xlib
Convert KeySym code to the name of the KeySym	<code>char *XKeysymToString(<i>keysym</i>)</code>	Xlib
Convert KeySym to the appropriate KeyCode	<code>KeyCode XKeysymToKeycode(<i>display, keysym</i>)</code>	Xlib
Convert name of the KeySym to the KeySym code	<code>Keysym XStringToKeysym(<i>string</i>)</code>	Xlib
Convert quark to a string	<code>char *XrmQuarkToString(<i>quark</i>)</code>	Xlib
Convert string to a quark	<code>XrmQuark XrmStringToQuark(<i>string</i>)</code>	Xlib

Description	Function	Manual
Convert string with one or more components to a binding list and a quark list	<code>XrmStringToBindingQuarkList(string, bindings_return, quarks_return)</code>	Xlib
Convert string with one or more components to a quark list	<code>void XrmStringToQuarkList(string, quarks_return)</code>	Xlib
Copy area between drawables of the same root and depth	<code>XCopyArea(display, src, desc, gc, src_x, src_y, width, height, dest_x, dest_y)</code>	Xlib
Copy components from a source graphics context to a destination graphics context	<code>XCopyGC(display, src, valuemask_copy, dest)</code>	Xlib
Copy contents of a rectangle on the display to a location with a pre-existing image structure	<code>XImage *XGetSubImage(display, d, x, y, width, height, plane_mask, format, dest_image, dest_x, dest_y)</code>	Xlib
Copy image stored in a file into a pixmap	<code>int XHPFileToPixmap(display, pixmap, cmap, gc, src_x, src_y, dst_x, dst_y, width, height, filename)</code>	Xlib
Copy image stored in a file into a window	<code>int XHPFileToWindow(display, w, modify_cmap, gc, src_x, src_y, dst_x, dst_y, width, height, filename)</code>	Xlib
Copy instance of a string	<code>String XtNewString(str)</code>	Intrinsics
Copy single bit-plane of a given drawable	<code>XCopyPlane(display, src, dest, gc, src_x, src_y, width, height, dest_x, dest_y, plane)</code>	Xlib
Create and manage a widget in a single procedure	<code>Widget XtCreateManagedWidget(name, widget_class, parent, args, num_args)</code>	Intrinsics
Create application context	<code>void XtCreateApplicationContext()</code>	Intrinsics
Create application shell instance	<code>Widget XtAppCreateShell(application_name, application_class, widget_class, display, args, num_args)</code>	Intrinsics

<b>Description</b>	<b>Function</b>	<b>Manual</b>
Create callback procedure that is called when there are file events	<code>void <i>InputCallbackProc</i>(client_data, source, id)</code>	Intrinsics
Create callback procedure that is called when timeouts expire	<code>void <i>TimerCallbackProc</i>(client_data, id)</code>	Intrinsics
Create color map for a screen	<code>Colormap <i>XCreateColormap</i>(display, w, visual, alloc)</code>	Xlib
Create cursor from a standard font	<code>Cursor <i>XCreateFontCursor</i>(display, shape)</code>	Xlib
Create cursor from font glyphs	<code>Cursor <i>XCreateGlyphCursor</i>(display, source_font, mask_font, source_char, mask_char, foreground_color, background_color)</code>	Xlib
Create cursor from two bitmaps	<code>Cursor <i>XCreatePixmapCursor</i>(display, source, mask, foreground_color, background_color)</code>	Xlib
Create entry in a given XAssocTable	<code>XMakeAssoc(display, table, x_id, data)</code>	Xlib
Create error handler	<code>void <i>ErrorHandler</i>(message)</code>	Intrinsics
Create event handler	<code>void <i>EventHandler</i>(w, client_data, event)</code>	Intrinsics
Create expose procedure	<code>void <i>ExposeProc</i>(w, event, region)</code>	Intrinsics
Create geometry manager for a composite widget class	<code>XtGeometryResult <i>GeometryHandler</i>(w, request, preferred_return)</code>	Intrinsics
Create instance of a widget	<code>XtCreateWidget(name, widget_class, parent, args, num_args)</code>	Intrinsics
Create instance of an application shell widget	<code>Widget <i>XtCreateApplicationShell</i>(name, widget_class, args, num_args)</code>	Intrinsics



Description	Function	Manual
Create new color map when allocating out of a previously shared color map has failed due to resource exhaustion	<code>Colormap XCopyColormapAndFree(<i>display</i>, <i>cmap</i>)</code>	Xlib
Create new graphics context that is usable on a screen with a depth of drawable	<code>GC XCreateGC(<i>display</i>, <i>d</i>, <i>valuemask_create</i>, <i>values</i>)</code>	Xlib
Create new, empty region	<code>Region XCreateRegion()</code>	Xlib
Create panels widget	<code>XwPanelWidgetClass</code>	Widgets
Create pixmap and then store bitmap-format data into it	<code>Pixmap XCreatePixmapFromBitmapData(<i>display</i>, <i>d</i>, <i>data</i>, <i>width</i>, <i>height</i>, <i>fg</i>, <i>bg</i>, <i>depth</i>)</code>	Xlib
Create pixmap of a given size	<code>Pixmap XCreatePixmap(<i>display</i>, <i>d</i>, <i>width</i>, <i>height</i>, <i>depth</i>)</code>	Xlib
Create pop-up shell	<code>Widget XtCreatePopupShell(<i>name</i>, <i>widget_class</i>, <i>parent</i>, <i>args</i>, <i>num_args</i>)</code>	Intrinsics
Create procedure to destroy widgets	<code>void WidgetProc(<i>w</i>)</code>	Intrinsics
Create procedure to realize a widget class	<code>void RealizeProc(<i>w</i>, <i>value_mask</i>, <i>attributes</i>)</code>	Intrinsics
Create procedure to set the insertion position of widget's children	<code>Cardinal OrderProc(<i>w</i>)</code>	Intrinsics
Create query geometry procedure	<code>XtGeometryResult QueryGeometry(<i>w</i>, <i>request</i>, <i>preferred_return</i>)</code>	Intrinsics
Create resource converter	<code>void Converter(<i>args</i>, <i>num_args</i>, <i>from</i>, <i>to</i>)</code>	Intrinsics
Create resource database from a string	<code>XrmDatabase XrmGetStringDatabase(<i>string</i>)</code>	Xlib
Create subimage	<code>XImage *XSubImage(<i>ximage</i>, <i>x</i>, <i>y</i>, <i>subimage_width</i>, <i>subimage_height</i>)</code>	Xlib

<b>Description</b>	<b>Function</b>	<b>Manual</b>
Create tile suitable for area filling or patterned text	<b>XwCreateTile</b>	Widgets
Create timeout in the default application context	<b>XtIntervalId XtAddTimeOut(<i>interval</i>, <i>proc</i>, <i>client_data</i>)</b>	Intrinsics
Create timeout value	<b>XtIntervalId XtAppAddTimeOut(<i>app_context</i>, <i>interval</i>, <i>proc</i>, <i>client_data</i>)</b>	Intrinsics
Create titlebar	<b>XwttitlebarWidgetClass</b>	Widgets
Create top level widget that is the root of another widget tree	<b>Widget XtCreateApplicationShell(<i>name</i>, <i>widget_class</i>, <i>num_args</i>)</b>	Intrinsics
Create union of a source region and a rectangle	<b>XUnionRectWithRegion(<i>rectangle</i>, <i>src_region</i>, <i>dest_region</i>-return)</b>	Xlib
Create unmapped InputOutput subwindow of a given parent	<b>Window XCreateSimpleWindow(<i>display</i>, <i>parent</i>, <i>x</i>, <i>y</i>, <i>width</i>, <i>height</i>, <i>border_width</i>, <i>border</i>, <i>background</i>)</b>	Xlib
Create unmapped window and set its attributes	<b>Window XCreateWindow(<i>display</i>, <i>parent</i>, <i>x</i>, <i>y</i>, <i>width</i>, <i>height</i>, <i>border_width</i>, <i>depth</i>, <i>class</i>, <i>visual</i>, <i>valuemask</i>, <i>attributes</i>)</b>	Xlib
Create windows for widgets	<b>void XtCreateWindow(<i>w</i>, <i>window_class</i>, <i>visual</i>, <i>value_mask</i>, <i>attributes</i>)</b>	Intrinsics
Deallocate a graphics context	<b>void XtDestroyGC(<i>gc</i>)</b>	Intrinsics
Deallocate memory allocated in a previous call to XCreateImage	<b>int XDestroyImage(<i>ximage</i>)</b>	Xlib
Deallocate storage associated with a specified region	<b>XDestroyRegion(<i>r</i>)</b>	Xlib
Define a pushbutton widget	<b>XwpushButtonWidgetClass</b>	Widgets
Define button meta class	<b>XwbuttonWidgetClass</b>	Widgets

<b>Description</b>	<b>Function</b>	<b>Manual</b>
Define menu manager meta widget	<code>XwmenumgrWidgetClass</code>	Widgets
Define menubutton widget	<code>XwmenubuttonWidgetClass</code>	Widgets
Define menupane meta widget	<code>XwmenupaneWidgetClass</code>	Widgets
Define popup and pulldown menupane widget	<code>XwcascadeWidgetClass</code>	Widgets
Define primitive widget meta class	<code>XwprimitiveWidgetClass</code>	Widgets
Define scrollbar widget	<code>XwscrollbarWidgetClass</code>	Widgets
Define scrolled window widget	<code>XswindowWidgetClass</code>	Widgets
Define static image widget	<code>XwstaticrasterWidgetClass</code>	Widgets
Define valuator widget	<code>XwvaluatorWidgetClass</code>	Widgets
Define which cursor will be used in a window	<code>XDefineCursor(<i>display</i>, <i>w</i>, <i>cursor</i>)</code>	Xlib
Delete context entry for a given window and type	<code>int XDeleteContext(<i>display</i>, <i>w</i>, <i>context</i>)</code>	Xlib
Delete entry from a given XAssocTable	<code>XDeleteAssoc(<i>display</i>, <i>table</i>, <i>x_id</i>)</code>	Xlib
Delete entry from an XModifierKeymap structure	<code>XModifierKeymap *XDeleteModifiermapEntry(<i>modmap</i>, <i>keycode_entry</i>, <i>modifier</i>)</code>	Xlib
Delete property on a window	<code>XDeleteProperty(<i>display</i>, <i>w</i>, <i>property</i>)</code>	Xlib
Describes interface to constraint resources	<code>Constraint</code>	Widgets
Destroy all subwindows of a window	<code>XDestroySubwindows(<i>display</i>, <i>w</i>)</code>	Xlib



Description	Function	Manual
Destroy application context	<code>void XtDestroyApplicationContext(<i>app_context</i>)</code>	Intrinsics
Destroy client	<code>XKillClient(<i>display</i>, <i>resource</i>)</code>	Xlib
Destroy colormap	<code>XFreeColormap(<i>display</i>, <i>cmap</i>)</code>	Xlib
Destroy widget and its children	<code>void XtDestroyWidget(<i>w</i>)</code>	Intrinsics
Destroy window and all of its subwindows	<code>XDestroyWindow(<i>display</i>, <i>w</i>)</code>	Xlib
Destroy windows associated with a widget	<code>void XtUnrealizeWidget(<i>w</i>)</code>	Intrinsics
Destroy XModifierKeymap structure	<code>XFreeModifiermap(<i>modmap</i>)</code>	Xlib
Determine bounding box of a 2-byte character string in a given font	<code>XTextExtents16(<i>font_struct</i>, <i>string</i>, <i>nchars</i>, <i>direction_return</i>, <i>font_ascent_return</i>, <i>font_descent_return</i>, <i>overall_return</i>)</code>	Xlib
Determine bounding box of an 8-bit character string in a given font	<code>XTextExtents(<i>font_struct</i>, <i>string</i>, <i>nchars</i>, <i>direction_return</i>, <i>font_ascent_return</i>, <i>font_descent_return</i>, <i>overall_return</i>)</code>	Xlib
Determine byte offset of a field within a structure	<code>Cardinal XtOffset(<i>pointer_type</i>, <i>field_name</i>)</code>	Intrinsics
Determine number of elements in a fixed-size array	<code>Cardinal XtNumber(<i>array</i>)</code>	Intrinsics
Determine upper and lower case equivalents for a KeySym	<code>void XtConvertCase(<i>dpy</i>, <i>keysym</i>, <i>lower_return</i>, <i>upper_return</i>)</code>	Intrinsics
Determine useful cursor sizes	<code>Status XQueryBestCursor(<i>display</i>, <i>d</i>, <i>width</i>, <i>height</i>, <i>width_return</i>, <i>height_return</i>)</code>	Xlib
Determine width of a two-byte character string	<code>int XTextWidth16(<i>font_struct</i>, <i>string</i>, <i>count</i>)</code>	Xlib
Determine width of an 8-bit character string	<code>int XTextWidth(<i>font_struct</i>, <i>string</i>, <i>count</i>)</code>	Xlib

<b>Description</b>	<b>Function</b>	<b>Manual</b>
Disable access control	<code>XDisableAccessControl(<i>display</i>)</code>	Xlib
Disable reset key sequence	<code>XHPDisableReset(<i>display</i>)</code>	Xlib
Discontinue a source of input	<code>void XtRemoveInput(<i>id</i>)</code>	Intrinsics
Display static text widget	<code>XwstatictextWidgetClass</code>	Widgets
Draw 2-byte characters in a given drawable	<code>XDrawString16(<i>display</i>, <i>d</i>, <i>gc</i>, <i>x</i>, <i>y</i>, <i>string</i>, <i>length</i>)</code>	Xlib
Draw 2-byte characters in a given drawable	<code>XDrawText16(<i>display</i>, <i>d</i>, <i>gc</i>, <i>x</i>, <i>y</i>, <i>items</i>, <i>nitems</i>)</code>	Xlib
Draw 2-byte image text characters in a given drawable	<code>XDrawImageString16(<i>display</i>, <i>d</i>, <i>gc</i>, <i>x</i>, <i>y</i>, <i>string</i>, <i>length</i>)</code>	Xlib
Draw 8-bit characters in a given drawable	<code>XDrawString(<i>display</i>, <i>d</i>, <i>gc</i>, <i>x</i>, <i>y</i>, <i>string</i>, <i>length</i>)</code>	Xlib
Draw 8-bit characters in a given drawable	<code>XDrawText(<i>display</i>, <i>d</i>, <i>gc</i>, <i>x</i>, <i>y</i>, <i>items</i>, <i>nitems</i>)</code>	Xlib
Draw 8-bit image text characters in a given drawable	<code>XDrawImageString(<i>display</i>, <i>d</i>, <i>gc</i>, <i>x</i>, <i>y</i>, <i>string</i>, <i>length</i>)</code>	Xlib
Draw arc in a given drawable	<code>XDrawArc(<i>display</i>, <i>d</i>, <i>gc</i>, <i>x</i>, <i>y</i>, <i>width</i>, <i>height</i>, <i>angle1</i>, <i>angle2</i>)</code>	Xlib
Draw arrow widget	<code>XwarrowWidgetClass</code>	Widgets
Draw line between two points in a given drawable	<code>XDrawLine(<i>display</i>, <i>d</i>, <i>gc</i>, <i>x1</i>, <i>y1</i>, <i>x2</i>, <i>y2</i>)</code>	Xlib
Draw multiple arcs in a given drawable	<code>XDrawArcs(<i>display</i>, <i>d</i>, <i>gc</i>, <i>arcs</i>, <i>narcs</i>)</code>	Xlib
Draw multiple lines in a given drawable	<code>XDrawLines(<i>display</i>, <i>d</i>, <i>gc</i>, <i>points</i>, <i>npoints</i>, <i>mode</i>)</code>	Xlib
Draw multiple points in a given drawable	<code>XDrawPoints(<i>display</i>, <i>d</i>, <i>gc</i>, <i>points</i>, <i>npoints</i>, <i>mode</i>)</code>	Xlib

Description	Function	Manual
Draw multiple, unconnected lines in a given drawable	<code>XDrawSegments(<i>display</i>, <i>d</i>, <i>gc</i>, <i>segments</i>, <i>nsegments</i>)</code>	Xlib
Draw outline of a single rectangle in a given drawable	<code>XDrawRectangle(<i>display</i>, <i>d</i>, <i>g</i>, <i>x</i>, <i>y</i>, <i>width</i>, <i>height</i>)</code>	Xlib
Draw outline of multiple rectangles in a given drawable	<code>XDrawRectangles(<i>display</i>, <i>d</i>, <i>gc</i>, <i>rectangles</i>, <i>nrectangles</i>)</code>	Xlib
Draw point in a given drawable	<code>XDrawPoint(<i>display</i>, <i>d</i>, <i>gc</i>, <i>x</i>, <i>y</i>)</code>	Xlib
Enable access control	<code>XEnableAccessControl(<i>display</i>)</code>	Xlib
Enable or disable synchronization	<code>int (*XSynchronize(<i>display</i>, <i>onoff</i>))()</code>	Xlib
Enable the reset key sequence	<code>XHPEnableReset(<i>display</i>)</code>	Xlib
Execute procedures in a callback list	<code>void XtCallCallbacks(<i>w</i>, <i>callback_name</i>, <i>call_data</i>)</code>	Intrinsics
Exit application	<code>XCLOSEDisplay</code>	Intrinsics
Fill arc in a given drawable	<code>XFillArc(<i>display</i>, <i>d</i>, <i>gc</i>, <i>x</i>, <i>y</i>, <i>width</i>, <i>height</i>, <i>angle1</i>, <i>angle2</i>)</code>	Xlib
Fill multiple arcs in a given drawable	<code>XFillArcs(<i>display</i>, <i>d</i>, <i>gc</i>, <i>arcs</i>, <i>narcs</i>)</code>	Xlib
Fill multiple rectangles in a given drawable	<code>XFillRectangles(<i>display</i>, <i>d</i>, <i>gc</i>, <i>rectangles</i>, <i>nrectangles</i>)</code>	Xlib
Fill polygon area in a given drawable	<code>XFillPolygon(<i>display</i>, <i>d</i>, <i>gc</i>, <i>points</i>, <i>npoints</i>, <i>shape</i>, <i>mode</i>)</code>	Xlib
Fill rectangle in a given drawable	<code>XFillRectangle(<i>display</i>, <i>d</i>, <i>gc</i>, <i>x</i>, <i>y</i>, <i>width</i>, <i>height</i>)</code>	Xlib
Flush GC cache, if any	<code>FlushGC(<i>display</i>, <i>gc</i>)</code>	Xlib
Flush output buffer	<code>XFlush(<i>display</i>)</code>	Xlib



<b>Description</b>	<b>Function</b>	<b>Manual</b>
Flush output buffer and wait until all requests have been processed	<code>XSync(<i>display</i>, <i>discard</i>)</code>	Xlib
Force the screen saver on or off	<code>XForceScreenSaver(<i>display</i>, <i>mode</i>)</code>	Xlib
Frame widget	<code>XwframeWidgetClass</code>	Widgets
Free all storage associated with a pixmap	<code>XFreePixmap(<i>display</i>, <i>pixmap</i>)</code>	Xlib
Free allocated block of storage	<code>void XtFree(<i>ptr</i>)</code>	Intrinsics
Free color map cells	<code>XFreeColors(<i>display</i>, <i>cmap</i>, <i>pixels</i>, <i>npixels</i>, <i>planes</i>)</code>	Xlib
Free cursor	<code>XFreeCursor(<i>display</i>, <i>cursor</i>)</code>	Xlib
Free data returned by XGetFontPath	<code>XFreeFontPath(<i>list</i>)</code>	Xlib
Free font information	<code>XFreeFontInfo(<i>names</i>, <i>free_info</i>, <i>actual_count</i>)</code>	Xlib
Free font name array	<code>XFreeFontName(<i>list</i>)</code>	Xlib
Free given graphics context	<code>XFreeGC(<i>display</i>, <i>gc</i>)</code>	Xlib
Free in-memory data created by an Xlib function	<code>XFree(<i>data</i>)</code>	Xlib
Free memory associated with a given XAssocTable	<code>XDestroyAssocTable(<i>table</i>)</code>	Xlib
Free XHPDeviceList	<code>void XHPFreeDeviceList(<i>list</i>)</code>	Xlib
Generate region from a polygon	<code>Region XPolygonRegion(<i>points</i>, <i>n</i>, <i>fill_rule</i>)</code>	Xlib
Generate smallest rectangle enclosing the region	<code>XClipBox(<i>r</i>, <i>rect_return</i>)</code>	Xlib

<b>Description</b>	<b>Function</b>	<b>Manual</b>
Get "black" pixel value	<code>unsigned long BlackPixel(<i>display</i>, <i>screen_number</i>)</code>	Xlib
Get "black" pixel value	<code>unsigned long XBlackPixel(<i>display</i>, <i>screen_number</i>)</code>	Xlib
Get "black" pixel value of screen	<code>unsigned long XBlackPixelOfScreen(<i>screen</i>)</code>	Xlib
Get "white" pixel value	<code>unsigned long XWhitePixel(<i>display</i>, <i>screen_number</i>)</code>	Xlib
Get "white" pixel value of screen	<code>unsigned long XWhitePixelOfScreen(<i>screen</i>)</code>	Xlib
Get and remove the next event in the queue that matches an event type	<code>Bool XCheckTypedEvent(<i>display</i>, <i>event_type</i>, <i>event_return</i>)</code>	Xlib
Get and remove the next event in the queue that matches an event type and a window	<code>Bool XCheckTypedWindowEvent(<i>display</i>, <i>w</i>, <i>event_type</i>, <i>event_return</i>)</code>	Xlib
Get and remove the next event that matches an event mask	<code>Bool XCheckMaskEvent(<i>display</i>, <i>event_mask</i>, <i>event_return</i>)</code>	Xlib
Get application context for a widget	<code>XtAppContext XtWidgetToApplicationContext(<i>w</i>)</code>	Intrinsics
Get atom for a given name	<code>Atom XInternalAtom(<i>display</i>, <i>atom_name</i>, <i>only_if_exists</i>)</code>	Xlib
Get best fill tile shape	<code>Status XQueryBestTile(<i>display</i>, <i>which_screen</i>, <i>width</i>, <i>height</i>, <i>width_return</i>, <i>height_return</i>)</code>	Xlib
Get best size of a tile, stipple, or cursor	<code>Status XQueryBestSize(<i>display</i>, <i>class</i>, <i>which_screen</i>, <i>width</i>, <i>height</i>, <i>width_return</i>, <i>height_return</i>)</code>	Xlib

<b>Description</b>	<b>Function</b>	<b>Manual</b>
Get best stipple shape	Status XQueryBestStipple( <i>display, which_screen, width, height, width_return, height_return</i> )	Xlib
Get bit vector that describes the state of the keyboard	XQueryKeymap( <i>display, keys_return</i> )	Xlib
Get bounding box of a 2-byte character string in a given font	XQueryTextExtent16( <i>display, font_ID, string, nchars, direction_return, font_ascent_return, font_descent_return, overall_return</i> )	Xlib
Get bounding box of an 8-bit character string in a given font	XQueryTextExtents( <i>display, font_ID, string, nchars, direction_return, font_ascent_return, font_descent_return, overall_return</i> )	Xlib
Get class of a widget	XtClass( <i>w</i> )	Intrinsics
Get class of a window	Status XGetClassHint( <i>display, w, class_hints_return</i> )	Xlib
Get connection number	int XConnectionNumber( <i>display</i> )	Xlib
Get contents of a rectangle in a given drawable on the display	XImage *XGetImage( <i>display, d, x, y, width, height, plane_mask, format</i> )	Xlib
Get control attributes of an extension input device	XHPGetDeviceControl( <i>display, deviceid, values_return</i> )	Xlib
Get current attributes of a window	Status XGetWindowAttributes( <i>display, w, window_attributes_return</i> )	Xlib
Get current control values for the keyboard	XGetKeyboardControl( <i>display, values_return</i> )	Xlib
Get current event selection mask for an extended input device	XHPGetCurrentDeviceMask( <i>display, window, deviceid, mask_return</i> )	Xlib
Get current font search path	char **XGetFontPath( <i>display, npaths_return</i> )	Xlib



Description	Function	Manual
Get current geometry of a given drawable	Status XGetGeometry( <i>display</i> , <i>d</i> , <i>root_return</i> , <i>x_return</i> , <i>y_return</i> , <i>width_return</i> , <i>height_return</i> , <i>border_width_return</i> , <i>depth_return</i> )	Xlib
Get current input focus	XGetInputFocus( <i>display</i> , <i>focus_return</i> , <i>revert_to_return</i> )	Xlib
Get current pointer parameters	XGetPointerParameters( <i>display</i> , <i>accel_numerator_return</i> , <i>accel_denominator_return</i> , <i>threshold_return</i> )	Xlib
Get current screen saver values	XGetScreenSaver( <i>display</i> , <i>timeout_return</i> , <i>interval_return</i> , <i>prefer_blanking_return</i> , <i>allow_exposures_return</i> )	Xlib
Get data associated with a window and type	int XFindContext( <i>display</i> , <i>w</i> , <i>context</i> , <i>data_return</i> )	Xlib
Get data from a given XAssocTable	char *XLookupAssoc( <i>display</i> , <i>table</i> , <i>x_id</i> )	Xlib
Get data from a specified cut buffer	char *XFetchBuffer( <i>display</i> , <i>nbytes_return</i> , <i>buffer</i> )	Xlib
Get data from cut buffer 0	char *XFetchBytes( <i>display</i> , <i>nbytes_return</i> )	Xlib
Get default color map of screen	Colormap XDefaultColormapOfScreen( <i>screen</i> )	Xlib
Get default colormap ID for allocation on the screen	Colormap XDefaultColormap( <i>display</i> , <i>screen_number</i> )	Xlib
Get default depth of the root window of the screen	int XDefaultDepthOfScreen( <i>screen</i> )	Xlib
Get default graphics context for the root window	GC XDefaultGC( <i>display</i> , <i>screen_number</i> )	Xlib
Get default graphics context of screen	GC XDefaultGCOfScreen( <i>screen</i> )	Xlib
Get default screen	Screen *XDefaultScreenOfDisplay( <i>display</i> )	Xlib

<b>Description</b>	<b>Function</b>	<b>Manual</b>
Get default screen number referenced in XOpenDisplay	<code>int XDefaultScreen(<i>display</i>)</code>	Xlib
Get default visual of screen	<code>Visual *XDefaultVisualOfScreen(<i>screen</i>)</code>	Xlib
Get depth of default root window	<code>int XDefaultDepth(<i>display</i>, <i>screen_number</i>)</code>	Xlib
Get depth of root window of specified screen	<code>int XDisplayPlanes(<i>display</i>, <i>screen_number</i>)</code>	Xlib
Get display of screen	<code>Display *XDisplayOfScreen(<i>screen</i>)</code>	Xlib
Get error messages from the error data base	<code>XGetErrorDatabaseText(<i>display</i>, <i>name</i>, <i>message</i>, <i>default_string</i>, <i>buffer_return</i>, <i>length</i>)</code>	Xlib
Get event mask and event type for a specific extended input event	<code>int XHPGetExtEventMask(<i>display</i>, <i>event_constant</i>, <i>event_type</i>, <i>mask</i>)</code>	Xlib
Get event mask of root window of the screen at connection setup time	<code>long XEventMaskOfScreen(<i>screen</i>)</code>	Xlib
Get first extension data structure for the extension numbered number	<code>XExtData *XFindOnExtensionList(<i>structure</i>, <i>number</i>)</code>	Xlib
Get focus window id and current focus state of an extended input device	<code>XHPGetDeviceFocus(<i>display</i>, <i>deviceid</i>, <i>focus_return</i>, <i>revert_to_return</i>)</code>	Xlib
Get font property	<code>Bool XGetFontProperty(<i>font_struct</i>, <i>atom</i>, <i>value_return</i>)</code>	Xlib
Get GContext resource ID	<code>GContext XGContextFromGC(<i>gc</i>)</code>	Xlib
Get height of screen in millimeters	<code>int XDisplayHeightMM(<i>display</i>, <i>screen_number</i>)</code>	Xlib
Get height of screen in millimeters	<code>int XHeightMMOfScreen(<i>screen</i>)</code>	Xlib

Description	Function	Manual
Get height of screen in pixels	<code>int XDisplayHeight(<i>display</i>, <i>screen_number</i>)</code>	Xlib
Get height of screen in pixels	<code>int XHeightOfScreen(<i>screen</i>)</code>	Xlib
Get host list	<code>XHostAddress *XListHosts(<i>display</i>, <i>nhosts_return</i>, <i>state_return</i>)</code>	Xlib
Get icon sizes hints for a window	<code>Status XGetIconSizes(<i>display</i>, <i>w</i>, <i>size_list_return</i>, <i>count_return</i>)</code>	Xlib
Get ID of a widget's parent	<code>Widget XtParent(<i>w</i>)</code>	Intrinsics
Get information about a stored image	<code>int XHPQueryImageFile(<i>filename</i>, <i>xwd_header_return</i>)</code>	Xlib
Get information about an available font	<code>XFontStruct *XQueryFont(<i>display</i>, <i>font_ID</i>)</code>	Xlib
Get key mapping of an extension input device	<code>XHPGetDeviceKeyMapping(<i>display</i>, <i>deviceid</i>, <i>first_keycode_wanted</i>, <i>keycode_count</i>, <i>keysyms_per_keycode_return</i>)</code>	Xlib
Get keycodes that are used as modifiers	<code>XModifierKeymap *XGetModifiersMapping(<i>display</i>)</code>	Xlib
Get length of the event queue for the connect display	<code>int XQLength(<i>display</i>)</code>	Xlib
Get list of extensions supported by the server	<code>char **XListExtensions(<i>display</i>, <i>nextensions_return</i>)</code>	Xlib
Get list of resource database levels	<code>Bool XrmQGetSearchList(<i>database</i>, <i>names</i>, <i>classes</i>, <i>list_return</i>, <i>list_length</i>)</code>	Xlib
Get list of the available font names	<code>char **XListFonts(<i>display</i>, <i>pattern</i>, <i>maxnames</i>, <i>actual_count_return</i>)</code>	Xlib
Get list of the currently installed color maps	<code>Colormap *XListInstalledColormaps(<i>display</i>, <i>w</i>, <i>num_return</i>)</code>	Xlib



<b>Description</b>	<b>Function</b>	<b>Manual</b>
Get list of visual information structures that match a specified template	<code>XVisualInfo *XGetVisualInfo(<i>display</i>, <i>vinfo_mask</i>, <i>vinfo_template</i>, <i>nitems_return</i>)</code>	Xlib
Get major version number of the protocol associated with the connected display	<code>int XProtocolVersion(<i>display</i>)</code>	Xlib
Get maximum number of installed color maps supported by screen	<code>int XMaxCmapsOfScreen(<i>screen</i>)</code>	Xlib
Get minimum number of installed color maps supported by screen	<code>int XMinCmapsOfScreen(<i>screen</i>)</code>	Xlib
Get minor protocol revision number of the X server	<code>int XProtocolRevision(<i>display</i>)</code>	Xlib
Get mode of the specified screen	<code>XHPGetServerMode(<i>display</i>, <i>screen</i>)</code>	Xlib
Get modifier mapping of an extension input device	<code>XHPGetDeviceModiferMapping(<i>display</i>, <i>deviceid</i>)</code>	Xlib
Get motion history for a specified extended device, window, and time	<code>XHPTimeCoord *XHPGetDeviceMotionEvents(<i>display</i>, <i>w</i>, <i>deviceid</i>, <i>start</i>, <i>stop</i>, <i>nevents_return</i>)</code>	Xlib
Get motion history for a specified window and time	<code>XTimeCoord *XGetMotionEvents(<i>display</i>, <i>w</i>, <i>start</i>, <i>stop</i>, <i>nevents_return</i>)</code>	Xlib
Get name a window wants displayed in its icon	<code>Status XGetIconName(<i>display</i>, <i>w</i>, <i>icon_name_return</i>)</code>	Xlib
Get name for a given atom identifier	<code>char *XGetAtomName(<i>display</i>, <i>atom</i>)</code>	Xlib
Get name of a window	<code>Status XFetchName(<i>display</i>, <i>w</i>, <i>window_name_return</i>)</code>	Xlib
Get names and information about available fonts	<code>char **XListFontsWithInfo(<i>display</i>, <i>pattern</i>, <i>maxnames</i>, <i>count_return</i>, <i>info_return</i>)</code>	Xlib

<b>Description</b>	<b>Function</b>	<b>Manual</b>
Get next event and remove it from queue	<code>XNextEvent(<i>display</i>, <i>event_return</i>)</code>	Xlib
Get number of available screens	<code>int XScreenCount(<i>display</i>)</code>	Xlib
Get number of bits to pad each scanline	<code>int XBitmapPad(<i>display</i>)</code>	Xlib
Get number of color map cells in the default color map	<code>int XCellsOfScreen(<i>screen</i>)</code>	Xlib
Get number of entries in the default color map	<code>int XDisplayCells(<i>display</i>, <i>screen_number</i>)</code>	Xlib
Get number of events that are pending	<code>int XPending(<i>display</i>)</code>	Xlib
Get number of planes in the root window of the screen	<code>int XPlanesOfScreen(<i>screen</i>)</code>	Xlib
Get number related to a vendor's release of the X server	<code>int XVendorRelease(<i>display</i>)</code>	Xlib
Get parent, list of children, and number of children for a window	<code>Status XQueryTree(<i>display</i>, <i>w</i>, <i>root_return</i>, <i>parent_return</i>, <i>children_return</i>, <i>nchildren_return</i>)</code>	Xlib
Get pixel value in an image	<code>unsigned long XGetPixel(<i>ximage</i>, <i>x</i>, <i>y</i>)</code>	Xlib
Get pointer mapping	<code>int XGetPointerMapping(<i>display</i>, <i>map_return</i>, <i>nmap</i>)</code>	Xlib
Get pointer to a newly created AssocTable	<code>XAssocTable *XCreateAssocTable(<i>size</i>)</code>	Xlib
Get pointer to a string that provides some identification of the owner of the X server implementation	<code>char *XServerVendor(<i>display</i>)</code>	Xlib

<b>Description</b>	<b>Function</b>	<b>Manual</b>
Get pointer to screen of specified display	<code>Screen *XScreenOfDisplay(<i>display</i>, <i>screen_number</i>)</code>	Xlib
Get pointer to the program's display	<code>Display *XtDisplay(<i>w</i>)</code>	Intrinsics
Get pointer to the program's screen	<code>Screen *XtScreen(<i>w</i>)</code>	Intrinsics
Get resource database for a display	<code>XrmDatabase XtDatabase(<i>display</i>)</code>	Intrinsics
Get resource list structure	<code>void XtGetResourceList(<i>class</i>, <i>resources_return</i>, <i>num_resources_return</i>)</code>	Intrinsics
Get resources for widget parts that are not widgets themselves	<code>void XtGetSubresources(<i>w</i>, <i>base</i>, <i>name</i>, <i>class</i>, <i>resources</i>, <i>num_resources</i>, <i>args</i>, <i>num_args</i>)</code>	Intrinsics
Get RGB values of a single pixel value	<code>XQueryColor(<i>display</i>, <i>cmap</i>, <i>def_in_out</i>)</code>	Xlib
Get RGB values of an array of pixels stored in color structures	<code>XQueryColors(<i>display</i>, <i>cmap</i>, <i>defs_in_out</i>, <i>ncolors</i>)</code>	Xlib
Get root window	<code>Window XRootWindow(<i>display</i>, <i>screen_number</i>)</code>	Xlib
Get root window for the default screen	<code>Window XDefaultRootWindow(<i>display</i>)</code>	Xlib
Get root window of screen	<code>Window XRootWindowOfScreen(<i>screen</i>)</code>	Xlib
Get root window the pointer is currently on and the pointer coordinates relative to the root's origin	<code>Bool XQueryPointer(<i>display</i>, <i>w</i>, <i>root_return</i>, <i>child_return</i>, <i>root_x_return</i>, <i>root_y_return</i>, <i>win_x_return</i>, <i>win_y_return</i>, <i>mask_return</i>)</code>	Xlib
Get selection owner	<code>Window XGetSelectionOwner(<i>display</i>, <i>selection</i>)</code>	Xlib



Description	Function	Manual
Get serial number of the last request known by Xlib to have been processed by the X server.	int XLastKnownRequestProcessed( <i>display</i> )	Xlib
Get serial number that is to be used for the next request	int XNextRequest( <i>display</i> )	Xlib
Get shared graphics contexts	GC XtGetGC( <i>w</i> , <i>value_mask</i> , <i>values</i> )	Intrinsics
Get size hints for a window in its normal state	Status XGetNormalHints( <i>display</i> , <i>w</i> , <i>hints_return</i> )	
Get size of a bitmap's scanline unit in bits	int XBitmapUnit( <i>display</i> )	Xlib
Get standard colormap structure associated with the described atom	Status XGetStandardColormap( <i>display</i> , <i>w</i> , <i>cmap_return</i> , <i>property</i> )	Xlib
Get string that was passed to XOpenDisplay when the current display was opened	char *XDisplayString( <i>display</i> )	Xlib
Get structure of the associated font	XFontStruct *XHPGet16bitMixedFont( <i>font</i> )	Xlib
Get superclass of a widget	XtSuperclass( <i>w</i> )	Intrinsics
Get symbols for the specified KeyCodes	KeySym *XGetKeyboardMapping( <i>display</i> , <i>first_keycode_wanted</i> , <i>keycode_count</i> , <i>keysyms_per_keycode_return</i> )	Xlib
Get textual descriptions of the specified error code	XGetErrorText( <i>display</i> , <i>code</i> , <i>buffer_return</i> , <i>length</i> )	Xlib
Get the next event on the queue	Boolean XtPeekEvent( <i>event_return</i> )	Intrinsics
Get transient property	Status XGetTransientForHint( <i>display</i> , <i>w</i> , <i>prop_window_return</i> )	Xlib

<b>Description</b>	<b>Function</b>	<b>Manual</b>
Get type, format, and value of a property of a window	<code>int XGetWindowProperty(<i>display</i>, <i>w</i>, <i>property</i>, <i>long_offset</i>, <i>long_length</i>, <i>delete</i>, <i>req_type</i>, <i>actual_type_return</i>, <i>actual_format_return</i>, <i>nitems_return</i>, <i>bytes_after_return</i>, <i>prop_return</i>)</code>	Xlib
Get value from the head of the input queue	<code>void XtAppNextEvent(<i>app_context</i>, <i>event_return</i>)</code>	Intrinsics
Get value from the head of the input queue without removing input from the queue	<code>Boolean XtAppPeekEvent(<i>app_context</i>, <i>event_return</i>)</code>	Intrinsics
Get value of selection timeout	<code>void XtGetSelectionValue(<i>w</i>, <i>selection</i>, <i>target</i>, <i>callback</i>, <i>client_data</i>, <i>time</i>)</code>	Intrinsics
Get value with all bits set on suitable for use in a plane argument to a procedure	<code>unsigned long XAllPlanes()</code>	Xlib
Get visual information that matches the specified depth and class of the screen	<code>Status XMatchVisualInfo(<i>display</i>, <i>screen</i>, <i>depth</i>, <i>class</i>, <i>vinfo_return</i>)</code>	Xlib
Get visual type for the specified screen	<code>Visual *XDefaultVisual(<i>display</i>, <i>screen_number</i>)</code>	Xlib
Get whether screen supports backing stores	<code>int XDoesBackingStore(<i>screen</i>)</code>	Xlib
Get whether screen supports save unders	<code>Bool XDoesSaveUnders(<i>screen</i>)</code>	Xlib
Get whether the leftmost bit in bitmap is the least or most significant	<code>int XBitmapBitOrder(<i>display</i>)</code>	Xlib
Get widget's preferred geometry	<code>XtGeometryResult XtQueryGeometry(<i>w</i>, <i>intended</i>, <i>preferred_return</i>)</code>	Intrinsics
Get widget's window ID	<code>Window XtWindow(<i>w</i>)</code>	Intrinsics

Description	Function	Manual
Get width of screen in millimeters	<code>int XDisplayWidthMM(<i>display</i>, <i>screen_number</i>)</code>	Xlib
Get width of screen in millimeters	<code>int XWidthMMOfScreen(<i>screen</i>)</code>	Xlib
Get width of screen in pixels	<code>int XDisplayWidth(<i>display</i>, <i>screen_number</i>)</code>	Xlib
Get width of screen in pixels	<code>int XWidthOfScreen(<i>screen</i>)</code>	Xlib
Get window's property list	<code>Atom *XListProperties(<i>display</i>, <i>w</i>, <i>num_prop_return</i>)</code>	Xlib
Get X environment defaults	<code>char *XGetDefault(<i>display</i>, <i>program</i>, <i>options</i>)</code>	Xlib
Grab button (passive) on an extension input device	<code>XHPGrabDeviceButton(<i>display</i>, <i>deviceid</i>, <i>button</i>, <i>modifiers</i>, <i>grab_window</i>, <i>owner_events</i>, <i>event_mask</i>, <i>pointer_mode</i>, <i>device_mode</i>)</code>	Xlib
Grab extended input device	<code>XHPGrabDevice(<i>display</i>, <i>deviceid</i>, <i>grab_window</i>, <i>pointer_mode</i>, <i>device_mode</i>, <i>owner_events</i>, <i>time</i>)</code>	Xlib
Grab key (passive) of the keyboard	<code>XGrabKey(<i>display</i>, <i>keycode</i>, <i>modifiers</i>, <i>grab_window</i>, <i>owner_events</i>, <i>pointer_mode</i>, <i>keyboard_mode</i>)</code>	Xlib
Grab key (passive) on an extension input device	<code>XHPGrabDeviceKey(<i>display</i>, <i>deviceid</i>, <i>keycode</i>, <i>modifiers</i>, <i>grab_window</i>, <i>owner_events</i>, <i>pointer_mode</i>, <i>device_mode</i>)</code>	Xlib
Grab keyboard	<code>int XGrabKeyboard(<i>display</i>, <i>grab_window</i>, <i>owner_events</i>, <i>pointer_mode</i>, <i>keyboard_mode</i>, <i>time</i>)</code>	Xlib
Grab pointer	<code>int XGrabPointer(<i>display</i>, <i>grab_window</i>, <i>owner_events</i>, <i>event_mask</i>, <i>pointer_mode</i>, <i>keyboard_mode</i>, <i>confine_to</i>, <i>cursor</i>, <i>time</i>)</code>	Xlib



Description	Function	Manual
Grab pointer button	<code>XGrabButton(<i>display</i>, <i>button_grab</i>, <i>modifiers</i>, <i>grab_window</i>, <i>owner_events</i>, <i>event_mask</i>, <i>pointer_mode</i>, <i>keyboard_mode</i>, <i>confine_to</i>, <i>cursor</i>)</code>	Xlib
Grab server	<code>XGrabServer(<i>display</i>)</code>	Xlib
Handle fatal I/O errors	<code>XSetIOErrorHandler(<i>handlers</i>)</code>	Xlib
Image editor widget	<code>XwimageEditWidgetClass</code>	Widgets
Include a bitmap in a program directly	<code>Pixmap XCreateBitmapFromData(<i>display</i>, <i>d</i>, <i>data</i>, <i>width</i>, <i>height</i>)</code>	Xlib
Increment each pixel in the pixmap	<code>XAddPixel(<i>ximage</i>, <i>value</i>)</code>	Xlib
Initialize class part	<code>void ClassProc(<i>widgetClass</i>)</code>	Intrinsics
Initialize display and add it to application context	<code>void XtDisplayInitialize(<i>app_context</i>, <i>display</i>, <i>application_name</i>, <i>application_class</i>, <i>options</i>, <i>num_options</i>, <i>argc</i>, <i>argv</i>)</code>	Intrinsics
Initialize extension	<code>XExtCodes *XInitExtensions(<i>display</i>, <i>name</i>)</code>	Xlib
Initialize hook procedure	<code>void ArgsProc(<i>w</i>, <i>args</i>, <i>num_args</i>)</code>	Intrinsics
Initialize resource manager	<code>void XrmInitialize()</code>	Xlib
Initialize widget class	<code>void InitProc(<i>request</i>, <i>new</i>)</code>	Intrinsics
Install accelerators from a widget on another widget	<code>void XtInstallAccelerators(<i>destination</i>, <i>source</i>)</code>	Intrinsics
Install all accelerators from a widget onto one destination	<code>void XtInstallAllAccelerators(<i>destination</i>, <i>source</i>)</code>	Intrinsics
Install color map	<code>XInstallColormap(<i>display</i>, <i>cmap</i>)</code>	Xlib
Invoke resource conversions	<code>void XtConvert(<i>w</i>, <i>from_type</i>, <i>from</i>, <i>to_type</i>, <i>to_return</i>)</code>	Intrinsics

Description	Function	Manual
List available input devices	XHPDeviceList *XHPListInputDevices( <i>display, ndevices</i> )	Xlib
List manager widget	XwlistWidgetClass	Widgets
Load font	Font XLoadFont( <i>display, name</i> )	Xlib
Load resource database from a C command line	void XrmParseCommand( <i>db, table, table_count, name, argc_in_out, argv_in_out</i> )	Xlib
Lock display structure	LockDisplay( <i>display</i> )	Xlib
Look up the KeySyms	KeySym XLookupKeysym( <i>key_event, index</i> )	Xlib
Look up the name of a color	Status XLookupColor( <i>display, cmap, color_name, exact_def_return, screen_def_return</i> )	Xlib
Lower a window so that it does not obscure any sibling windows	XLowerWindow( <i>display, w</i> )	Xlib
Lower highest mapped child of a window that occludes another child	XCirculateSubwindowsDown( <i>display, w</i> )	Xlib
Make a general geometry manager request from a widget	XtGeometryResult XtMakeGeometryRequest( <i>w, request, reply_return</i> )	Intrinsics
Make a simple resize request from a widget	XtGeometryResult XtMakeResizeRequest( <i>w, width, height, width_return, height_return</i> )	Intrinsics
Manage bulletin board widget	XwbulletinWidgetClass	Widgets
Manage general widget layout	XwformWidgetClass	Widgets
Manage popup menu widget	XwpopupmgrWidgetClass	Widgets
Manage pulldown menu widget	XwpulldownWidgetClass	Widgets

<b>Description</b>	<b>Function</b>	<b>Manual</b>
Manage row/column widget	<code>XwrowColWidgetClass</code>	Widgets
Manage vertical paned manager widget	<code>XwvPanedWidgetClass</code>	Widgets
Map all subwindows for a window	<code>XMapSubwindows(<i>display</i>, <i>w</i>)</code>	Xlib
Map and raise a window	<code>XMapRaised(<i>display</i>, <i>w</i>)</code>	Xlib
Map key event to an ASCII string	<code>int XLookupString(<i>event_struct</i>, <i>buffer_return</i>, <i>bytes_buffer</i>, <i>keysym_return</i>, <i>status_in_out</i>)</code>	Xlib
Map pop-up from within an application	<code>void XtPopup(<i>popup_shell</i>, <i>grab_kind</i>)</code>	Intrinsics
Map pop-up with exclusive grab from a callback list	<code>void XtCallbackExclusive(<i>w</i>, <i>client_data</i>, <i>call_data</i>)</code>	Intrinsics
Map pop-up with nonexclusive grab from a callback list	<code>void XtCallbackNonexclusive(<i>w</i>, <i>client_data</i>, <i>call_data</i>)</code>	Intrinsics
Map pop-up without grab from a callback list	<code>void XtCallbackNone(<i>w</i>, <i>client_data</i>, <i>call_data</i>)</code>	Intrinsics
Map widget explicitly	<code>XtMapWidget(<i>w</i>)</code>	Intrinsics
Map window	<code>XMapWindow(<i>display</i>, <i>w</i>)</code>	Xlib
Merge contents of two resource databases	<code>void XrmMergeDatabases(<i>source_db</i>, <i>target_db</i>)</code>	Xlib
Merge Expose and GraphicsExpose into a region	<code>void XtAddExposureToRegion(<i>event</i>, <i>region</i>)</code>	Intrinsics
Merge new translations into an existing translation table	<code>void XtAugmentTranslations(<i>w</i>, <i>translations</i>)</code>	Intrinsics
Merge two ArgList structures	<code>ArgList XtMergeArgLists(<i>args1</i>, <i>num_args1</i>, <i>args2</i>, <i>num_args2</i>)</code>	Intrinsics



Description	Function	Manual
Modify current value of resource associated with widget instance	<code>void XtSetValues(<i>w</i>, <i>args</i>, <i>num_args</i>)</code>	Intrinsics
Move and resize the sibling widget of the child making the geometry request	<code>void XtConfigureWidget(<i>w</i>, <i>x</i>, <i>y</i>, <i>width</i>, <i>height</i>, <i>border_width</i>)</code>	Intrinsics
Move keyboard focus (and the pointer) to a new top-level widget	<code>XwMoveFocus</code>	Widgets
Move pointer to an arbitrary point in the screen	<code>XWarpPointer(<i>display</i>, <i>src_w</i>, <i>dest_w</i>, <i>src_x</i>, <i>src_y</i>, <i>src_width</i>, <i>src_height</i>, <i>dest_x</i>, <i>dest_y</i>)</code>	Xlib
Move region by a specified amount	<code>XOffsetRegion(<i>r</i>, <i>dx</i>, <i>dy</i>)</code>	Xlib
Move sibling widget of the child making the geometry request	<code>void XtMoveWidget(<i>w</i>, <i>x</i>, <i>y</i>)</code>	Intrinsics
Move window without changing its size	<code>XMoveWindow(<i>display</i>, <i>w</i>, <i>x</i>, <i>y</i>)</code>	Xlib
Name window	<code>XStoreName(<i>display</i>, <i>w</i>, <i>window_name</i>)</code>	Xlib
Open connection to the X server controlling a display	<code>Display *XOpenDisplay(<i>display_name</i>)</code>	Xlib
Open display from a string	<code>Display *XtOpenDisplay(<i>app_context</i>, <i>display_name</i>, <i>application_name</i>, <i>application_class</i>, <i>options</i>, <i>num_options</i>, <i>argc</i>, <i>argv</i>)</code>	Intrinsics
Open extended input device	<code>int XHPSetInputDevice(<i>display</i>, <i>deviceid</i>, <i>mode</i>)</code>	Xlib
Overwrite existing translations with new translations	<code>void XtOverrideTranslations(<i>w</i>, <i>translations</i>)</code>	Intrinsics
Parse accelerators	<code>XtAccelerators</code> <code>XtParseAcceleratorTable(<i>source</i>)</code>	Intrinsics

Description	Function	Manual
Parse color values	Status XParseColor( <i>display, cmap, spec, exact_def_return</i> )	Xlib
Parse standard window geometry strings	int XParseGeometry( <i>parsestring, x_return, y_return, width_return, height_return</i> )	Xlib
Parse window geometry given a user-specified position and a default position	int XGeometry( <i>display, screen, position, default_position, bwidth, fwidth, fheight, xadder, yadder, x_return, y_return, width_return, height_return</i> )	Xlib
Peek at the event queue	XPeekEvent( <i>display, event_return</i> )	Xlib
Perform direct conversion	void XtDirectConvert( <i>converter, args, num_args, from, to_return</i> )	Intrinsics
Perform XLoadFont and XQueryFont in a single operation	XFontStruct *XLoadQueryFont( <i>display, name</i> )	Xlib
Pop down a spring-loaded menu	void MenuPopdown( <i>shell_name</i> )	Intrinsics
Pop down pop-ups that were created with one of the callback routines	void XtCallbackPopdown( <i>w, client_data, call_data</i> )	Intrinsics
Pop up a menu when a pointer button is pressed or when the pointer is moved on to some window	void MenuPopup( <i>shell_name</i> )	Intrinsics
Process input	void XtAppMainLoop( <i>app_context</i> )	Intrinsics
Push event back to the top of the event queue	XPutBackEvent( <i>display, event</i> )	Xlib
Raise lowest-mapped child of a window that is partly or completely occluded by another child	XCirculateSubwindowsUp( <i>display, w</i> )	Xlib

Description	Function	Manual
Raise window so that no sibling window obscures it	<code>XRaiseWindow(<i>display</i>, <i>w</i>)</code>	Xlib
Read bitmap in from a file	<code>int XReadBitmapFile(<i>display</i>, <i>d</i>, <i>filename</i>, <i>width_return</i>, <i>height_return</i>, <i>bitmap_return</i>, <i>x_hot_return</i>, <i>y_hot_return</i>)</code>	Xlib
Read value of any property of type WM.SIZE_HINTS	<code>Status XGetSizeHints(<i>display</i>, <i>w</i>, <i>hints_return</i>, <i>property</i>)</code>	Xlib
Read window manager hints	<code>XWMHints *XGetWMHints(<i>display</i>, <i>w</i>)</code>	Xlib
Read zoom hints for a window	<code>Status XGetZoomHints(<i>display</i>, <i>w</i>, <i>zhints_return</i>)</code>	Xlib
Realize widget instance	<code>void XtRealizeWidget(<i>w</i>)</code>	Intrinsics
Rebind meaning of a KeySym for a client	<code>XRebindKeySym(<i>display</i>, <i>keysym</i>, <i>list</i>, <i>mod_count</i>, <i>string</i>, <i>bytes_string</i>)</code>	Xlib
Redirect keyboard input to a child of a composite widget without calling XtSetInputFocus	<code>XtSetKeyboardFocus(<i>w</i>, <i>descendent</i>)</code>	Intrinsics
Redirect user input to a modal widget	<code>void XtAddGrab(<i>w</i>, <i>exclusive</i>, <i>spring_loaded</i>)</code>	Intrinsics
Reduce a region by a specified amount	<code>XShrinkRegion(<i>r</i>, <i>dx</i>, <i>dy</i>)</code>	Xlib
Refresh stored modifier and keymap information	<code>XRefreshKeyboardMapping(<i>event_map</i>)</code>	Xlib
Register action table with the translation manager	<code>void XtAddActions(<i>actions</i>, <i>num_actions</i>)</code>	Intrinsics
Register all of the resource converters used by the X Widgets	<code>XwRegisterConverters</code>	Widgets
Register case converter	<code>XtRegisterCaseConverter(<i>dpy</i>, <i>proc</i>, <i>start</i>, <i>stop</i>)</code>	Intrinsics



<b>Description</b>	<b>Function</b>	<b>Manual</b>
Register event handler procedure with the dispatch mechanism	<code>void XtAddEventHandler(<i>w</i>, <i>event_mask</i>, <i>nonmaskable</i>, <i>proc</i>, <i>client_data</i>)</code>	Intrinsics
Register event handler procedure with the dispatch mechanism without causing the server to select for that event	<code>void XtAddRawEventHandler(<i>w</i>, <i>event_mask</i>, <i>nonmaskable</i>, <i>proc</i>, <i>client_data</i>)</code>	Intrinsics
Register key translator	<code>void XtSetKeyTranslator(<i>dpy</i>, <i>proc</i>)</code>	Intrinsics
Register new file for input	<code>XtInputId XtAppAddInput(<i>app_context</i>, <i>source</i>, <i>condition</i>, <i>proc</i>, <i>client_data</i>)</code>	Intrinsics
Register new resource converter	<code>void XtAddConverter(<i>from_type</i>, <i>to_type</i>, <i>converter</i>, <i>convert_args</i>, <i>num_args</i>)</code>	Intrinsics
Register new sort of events with the application context	<code>XtInputId XtAddInput(<i>source</i>, <i>condition</i>, <i>proc</i>, <i>client_data</i>)</code>	Intrinsics
Register procedure to be called on fatal error conditions	<code>void XtSetErrorHandler(<i>msg_handler</i>)</code>	Intrinsics
Register procedure to be called on non-fatal error conditions	<code>void XtSetWarningHandler(<i>msg_handler</i>)</code>	Intrinsics
Register work proc	<code>XtWorkProcId XtAppAddWorkProc(<i>app_context</i>, <i>proc</i>, <i>client_data</i>)</code>	Intrinsics
Release grab of an extension input device	<code>XHPUngrabDevice(<i>display</i>, <i>deviceid</i>, <i>time</i>)</code>	Xlib
Release passive grab of a button on an extension input device	<code>XHPUngrabDeviceButton(<i>display</i>, <i>deviceid</i>, <i>button</i>, <i>modifiers</i>)</code>	Xlib

<b>Description</b>	<b>Function</b>	<b>Manual</b>
Relinquish ownership of selection	<code>void XtDisownSelection(<i>w</i>, <i>selection</i>, <i>time</i>)</code>	Intrinsics
Remove all callback procedures from a callback list	<code>void XtRemoveAllCallbacks(<i>w</i>, <i>callback_name</i>)</code>	Intrinsics
Remove callback procedure from a callback list	<code>void XtRemoveCallback(<i>w</i>, <i>callback_name</i>, <i>callback</i>, <i>client_data</i>)</code>	Intrinsics
Remove children from parent's managed set	<code>void XtUnmanageChildren(<i>children</i>, <i>num_children</i>)</code>	Intrinsics
Remove existing translations	<code>void XtUninstallTranslations(<i>w</i>)</code>	Intrinsics
Remove list of callback procedures from a callback list	<code>void XtRemoveCallbacks(<i>w</i>, <i>callback_name</i>, <i>callback</i>)</code>	Intrinsics
Remove multiple hosts at one time	<code>XRemoveHosts(<i>display</i>, <i>hosts</i>, <i>num_hosts</i>)</code>	Xlib
Remove next event that matches an event mask	<code>XMaskEvent(<i>display</i>, <i>event_mask</i>, <i>event_return</i>)</code>	Xlib
Remove next event that matches both a window and an event mask	<code>XCheckWindowEvent(<i>display</i>, <i>w</i>, <i>event_mask</i>, <i>event_return</i>)</code>	Xlib
Remove next event that matches both a window and an event mask	<code>XWindowEvent(<i>display</i>, <i>w</i>, <i>event_mask</i>, <i>event_return</i>)</code>	Xlib
Remove previously registered event handler	<code>void XtRemoveEventHandler(<i>w</i>, <i>event_mask</i>, <i>nonmaskable</i>, <i>proc</i>, <i>client_data</i>)</code>	Intrinsics
Remove previously registered raw event handler	<code>void XtRemoveRawEventHandler(<i>w</i>, <i>event_mask</i>, <i>nonmaskable</i>, <i>proc</i>, <i>client_data</i>)</code>	Intrinsics
Remove redirection of user input to a modal widget	<code>void XtRemoveGrab(<i>w</i>)</code>	Intrinsics
Remove single child from parent's managed set	<code>void XtUnmanageChild(<i>child</i>)</code>	Intrinsics

<b>Description</b>	<b>Function</b>	<b>Manual</b>
Remove single host	<code>XRemoveHost(<i>display</i>, <i>host</i>)</code>	Xlib
Remove window from the client's save-set	<code>XRemoveFromSaveSet(<i>display</i>, <i>w</i>)</code>	Xlib
Report error to the user when the requested display does not exist	<code>char *XDisplay(<i>string</i>)</code>	Xlib
Request conversion of a selection	<code>XConvertSelection(<i>display</i>, <i>selection</i>, <i>target</i>, <i>property</i>, <i>requestor</i>, <i>time</i>)</code>	Xlib
Request for extension	<code>Bool XQueryExtension(<i>display</i>, <i>name</i>, <i>major_opcode_return</i>, <i>first_event_return</i>, <i>first_error_return</i>)</code>	Xlib
Reset screen saver	<code>XResetScreenSaver(<i>display</i>)</code>	Xlib
Resize sibling widget of the child making the geometry request	<code>void XtResizeWidget(<i>w</i>, <i>width</i>, <i>height</i>, <i>border_width</i>)</code>	Intrinsics
Resize widget	<code>void Resize(<i>w</i>)</code>	Intrinsics
Restack set of windows from top to bottom	<code>XRestackWindows(<i>display</i>, <i>windows</i>, <i>nwindows</i>)</code>	Xlib
Retrieve current value of a non-widget resource data associated with a widget instance	<code>void XtGetSubvalues(<i>base</i>, <i>resources</i>, <i>num_resources</i>, <i>args</i>, <i>num_args</i>)</code>	Intrinsics
Retrieve current value of resource associated with a widget instance	<code>void XtGetValues(<i>w</i>, <i>args</i>, <i>num_args</i>)</code>	Intrinsics
Retrieve event mask for a widget	<code>void XtBuildEventMask(<i>w</i>)</code>	Intrinsics
Retrieve resource database from storage	<code>XrmDatabase</code> <code>XrmGetFileDatabase(<i>filename</i>)</code>	Xlib



<b>Description</b>	<b>Function</b>	<b>Manual</b>
Retrieve resource from a resource database	<code>Bool XrmGetResource(database, str_name, str_class, str_type_return, str_value_return)</code>	Xlib
Retrieve resource from a resource database	<code>Bool XrmQGetResource(database, quark_name, quark_class, quark_type_return, value_return)</code>	Xlib
Retrieve resources that are not specific to a widget but apply to the overall application	<code>void XtGetApplicationResources(w, base, resources, num_resources, args, num_args)</code>	Intrinsics
Ring the bell	<code>XBell(display, percent)</code>	Xlib
Rotate cut buffers	<code>XRotateBuffers(display, rotate)</code>	Xlib
Rotate property list of a window	<code>XRotateWindowProperties(display, w, properties, num_prop, npositions)</code>	Xlib
Save contents of a rectangular pixmap in a file	<code>int XHPPixmapToFile(display, pixmap, color, x, y, width, height, plane_mask, format, filename)</code>	Xlib
Save contents of a rectangular window area in a file	<code>int XHPWindowToFile(display, w, x, y, width, height, format, filename)</code>	Xlib
Save data value that corresponds to a window and context type	<code>int XSaveContext(display, w, context, data)</code>	Xlib
Search resource database levels for a given resource	<code>Bool XrmQGetSearchResource(list, name, class, type, type_return, value_return)</code>	Xlib
Select events to be reported to the client application	<code>XSelectInput(display, w, event_mask)</code>	Xlib
Select input from an extended input device	<code>XHPSelectExtensionEvent(display, window, deviceid, mask)</code>	Xlib
Send Acknowledge to an extended input device	<code>XHPAcknowledge(display, deviceid, acknowledge)</code>	Xlib

Description	Function	Manual
Send event to a specified window	Status XSendEvent( <i>display, w, propagate, event_mask, event_send</i> )	Xlib
Send events to registered functions and widgets	void XtAppProcessEvent( <i>app_context, mask</i> )	Intrinsics
Send events to registered functions and widgets	void XtDispatchEvent( <i>event</i> )	Intrinsics
Send no operation protocol request to the X server	XNoOp( <i>display</i> )	Xlib
Send prompt to an extended input device	XHPPrompt( <i>display, deviceid, prompt</i> )	Xlib
Separate menu items widget	XwmenuSepWidgetClass	Widgets
Set "after" function	int (*XSetAfterFunction( <i>display, proc</i> ))()	Xlib
Set arc mode of a graphics context	XSetArcMode( <i>display, gc, arc_mode</i> )	Xlib
Set argument in an Arglist	XtSetArg( <i>arg, name, value</i> )	Intrinsics
Set background of a given graphics context	XSetBackground( <i>display, gc, background</i> )	Xlib
Set background of a window to a given pixel	XSetWindowBackground( <i>display, w, background_pixel</i> )	Xlib
Set background of a window to a given pixmap	XSetWindowBackgroundPixmap( <i>display, w, background_pixmap</i> )	Xlib
Set class of a window	XSetClassHint( <i>display, w, class_hints</i> )	Xlib
Set clip origin of a graphics context	XSetClipOrigin( <i>display, gc, clip_x_origin, clip_y_origin</i> )	Xlib
Set clip-mask of a graphics context to a region	XSetRegion( <i>display, gc, r</i> )	Xlib
Set clip-mask of a graphics context	XSetClipMask( <i>display, gc, pixmap</i> )	Xlib

<b>Description</b>	<b>Function</b>	<b>Manual</b>
Set clip_mask of a graphics context to the specified list of rectangles	<code>XSetClipRectangles(<i>display, gc, clip_x_origin, clip_y_origin, rectangles, n, ordering</i>)</code>	Xlib
Set color of a pixel to a named color	<code>XStoreNamedColor(<i>display, cmap, color, pixel, flags</i>)</code>	Xlib
Set colormap of a window	<code>XSetWindowColormap(<i>display, w, cmap</i>)</code>	Xlib
Set command property	<code>XSetCommand(<i>display, w, argv, argc</i>)</code>	Xlib
Set current font of a graphics context	<code>XSetFont(<i>display, gc, font</i>)</code>	Xlib
Set current value of non-widget resource associated with a widget instance	<code>void XtSetSubvalues(<i>base, resources, num_resources, args, num_args</i>)</code>	Intrinsics
Set dash_offset and dash_list for dashed lines of a graphics context	<code>XSetDashes(<i>display, gc, dash_offset, dash_list, n</i>)</code>	Xlib
Set display function in a graphics context	<code>XSetFunction(<i>display, gc, function</i>)</code>	Xlib
Set error handler	<code>XSetErrorHandler(<i>handler</i>)</code>	Xlib
Set fill rule of a given graphics context	<code>XSetFillRule(<i>display, gc, fill_rule</i>)</code>	Xlib
Set fill style of a graphics context	<code>XSetFillStyle(<i>display, gc, fill_style</i>)</code>	Xlib
Set fill tile of a graphics context	<code>XSetTile(<i>display, gc, tile</i>)</code>	Xlib
Set focus of an extended input device	<code>XHPSetDeviceFocus(<i>display, deviceid, focus, revert_to, time</i>)</code>	Xlib
Set font search path	<code>XSetFontPath(<i>display, directories, ndirs</i>)</code>	Xlib
Set foreground of a given graphics context	<code>XSetForeground(<i>display, gc, foreground</i>)</code>	Xlib



Description	Function	Manual
Set foreground, background, plane mask, and function components for a given graphics context	<code>XSetState(<i>display</i>, <i>gc</i>, <i>foreground</i>, <i>background</i>, <i>function</i>, <i>plane_mask</i>)</code>	Xlib
Set graphics-exposures flag of a graphics context	<code>XSetGraphicsExposures(<i>display</i>, <i>gc</i>, <i>graphics_exposures</i>)</code>	Xlib
Set icon size hints for a window	<code>XSetIconSizes(<i>display</i>, <i>w</i>, <i>size_list</i>, <i>count</i>)</code>	Xlib
Set input focus	<code>XSetInputFocus(<i>display</i>, <i>focus</i>, <i>revert_to</i>, <i>time</i>)</code>	Xlib
Set line drawing components of a graphics context	<code>XSetLineAttributes(<i>display</i>, <i>gc</i>, <i>line_width</i>, <i>line_style</i>, <i>cap_style</i>, <i>join_style</i>)</code>	Xlib
Set mapping of buttons on the pointer	<code>int XSetPointerMapping(<i>display</i>, <i>map</i>, <i>nmap</i>)</code>	Xlib
Set name to be displayed in a window's icon	<code>XSetIconName(<i>display</i>, <i>w</i>, <i>icon_name</i>)</code>	Xlib
Set pixel value in an image	<code>int XPutPixel(<i>ximage</i>, <i>x</i>, <i>y</i>, <i>pixel</i>)</code>	Xlib
Set plane mask of a graphics context	<code>XSetPlaneMask(<i>display</i>, <i>gc</i>, <i>plane_mask</i>)</code>	
Set screen saver	<code>XSetScreenSaver(<i>display</i>, <i>timeout</i>, <i>interval</i>, <i>prefer_blanking</i>, <i>allow_exposures</i>)</code>	Xlib
Set selection owner	<code>XSetSelectionOwner(<i>display</i>, <i>selection</i>, <i>owner</i>, <i>time</i>)</code>	Xlib
Set sensitivity state of a widget	<code>void XtSetSensitive(<i>w</i>, <i>sensitive</i>)</code>	Intrinsics
Set size hints	<code>void XSetNormalHints(<i>display</i>, <i>w</i>, <i>hints</i>)</code>	Xlib
Set standard color map	<code>void XSetStandardColormap(<i>display</i>, <i>w</i>, <i>cmap</i>, <i>property</i>)</code>	Xlib
Set stipple of a graphics context	<code>XSetStipple(<i>display</i>, <i>gc</i>, <i>stipple</i>)</code>	Xlib

Description	Function	Manual
Set subwindow mode of a graphics context	<code>XSetSubwindowMode(<i>display</i>, <i>gc</i>, <i>subwindow_mode</i>)</code>	Xlib
Set tile or stipple origin of a graphics context	<code>XSetTSOrigin(<i>display</i>, <i>gc</i>, <i>ts_x_origin</i>, <i>ts_y_origin</i>)</code>	Xlib
Set toolkit selection timeout	<code>void XtSetSelectionTimeout(<i>timeout</i>)</code>	Intrinsics
Set transient property	<code>XSetTransientForHint(<i>display</i>, <i>w</i>, <i>prop_window</i>)</code>	Xlib
Set value of any property of type WM_SIZE_HINTS	<code>XSetSizeHints(<i>display</i>, <i>w</i>, <i>hints</i>, <i>property</i>)</code>	Xlib
Set which KeyCodes are to be used as modifiers	<code>XSetModifierMapping(<i>display</i>, <i>modmap</i>)</code>	Xlib
Set window manager hints	<code>XSetWMHints(<i>display</i>, <i>w</i>, <i>wmhints</i>)</code>	Xlib
Set zoom hints for a window	<code>XSetZoomHints(<i>display</i>, <i>w</i>, <i>zhints</i>)</code>	Xlib
Specify minimum set of properties describing the "quickie" application	<code>XSetStandardProperties(<i>display</i>, <i>w</i>, <i>window_name</i>, <i>icon_name</i>, <i>icon_pixmap</i>, <i>argv</i>, <i>argc</i>, <i>hints</i>)</code>	Xlib
Specify required byte order for each scanline in XY format or for each pixel value in Z format	<code>int XImageByteOrder(<i>display</i>)</code>	Xlib
Start an Xt program	<code>Widget XtInitialize(<i>shell_name</i>, <i>application_class</i>, <i>options</i>, <i>num_options</i>, <i>argc</i>, <i>argv</i>)</code>	Intrinsics
Store copy of a resource database in nonvolatile storage	<code>void XrmPutFileDatabase(<i>database</i>, <i>stored_db</i>)</code>	Xlib
Store data in a specified cut buffer	<code>XStoreBuffer(<i>display</i>, <i>bytes</i>, <i>nbytes</i>, <i>buffer</i>)</code>	Xlib
Store data in cut buffer 0	<code>XStoreBytes(<i>display</i>, <i>bytes</i>, <i>nbytes</i>)</code>	Xlib
Store resources into the database	<code>void XrmPutResource(<i>database</i>, <i>specifier</i>, <i>type</i>, <i>value</i>)</code>	Xlib

Description	Function	Manual
Store resources into the database	<code>void XrmQPutResource(database, bindings, quarks, type, value)</code>	Xlib
Store RGB value into a single color map cell	<code>XStoreColor(display, cmap, color)</code>	Xlib
Store RGB values into color map cells	<code>XStoreColors(display, cmap, color, ncolors)</code>	Xlib
Subtract two regions	<code>XSubtractRegion(sra, srb, dr_return)</code>	Xlib
Test if a widget is a subclass of Composite	<code>void XtIsComposite(w)</code>	Intrinsics
Toggle button widget	<code>XwtoggleWidgetClass</code>	Widgets
Translate coordinates from one window to another	<code>Bool XTranslateCoordinates(display, src_w, dest_w, src_x, src_y, dest_x_return, dest_y_return, child_return)</code>	Xlib
Translate widget name to widget instance	<code>Widget XtNameToWidget(reference, names)</code>	Intrinsics
Translate window and display pointer into a widget instance	<code>Widget XtWindowToWidget(display, window)</code>	Intrinsics
Translate x-y coordinate pair from widget coordinates to root coordinates	<code>void XtTranslateCoords(w, x, y, rootx_return, rooty_return)</code>	Intrinsics
Turn off autorepeat for an extension input device	<code>XHPDeviceAutoRepeatOff(display, deviceid)</code>	Xlib
Turn off keyboard auto-repeat	<code>XAutoRepeatOff(display)</code>	Xlib
Turn on autorepeat for an extension input device	<code>XHPDeviceAutoRepeatOn(display, deviceid, mode)</code>	Xlib
Turn on keyboard auto-repeat	<code>XAutoRepeatOn(display)</code>	Xlib



Description	Function	Manual
Undefine the cursor in a given window	<code>XUndefineCursor(<i>display</i>, <i>w</i>)</code>	Xlib
Ungrab key	<code>XUngrabKey(<i>display</i>, <i>keycode</i>, <i>modifiers</i>, <i>ungrab_window</i>)</code>	Xlib
Ungrab keyboard	<code>XUngrabKeyboard(<i>display</i>, <i>time</i>)</code>	Xlib
Ungrab pointer	<code>XUngrabPointer(<i>display</i>, <i>time</i>)</code>	Xlib
Ungrab pointer button	<code>XUngrabButton(<i>display</i>, <i>button_ungrab</i>, <i>modifiers</i>, <i>ungrab_window</i>)</code>	Xlib
Ungrab server	<code>XUngrabServer(<i>display</i>)</code>	Xlib
Uninstall color map	<code>XUninstallColormap(<i>display</i>, <i>cmap</i>)</code>	Xlib
Unload font and free the storage used by the font structure	<code>XFreeFont(<i>display</i>, <i>font_struct</i>)</code>	Xlib
Unload font that was loaded by XLoadFont	<code>XUnloadFont(<i>display</i>, <i>font</i>)</code>	Xlib
Unlock display structure	<code>UnlockDisplay(<i>display</i>)</code>	Xlib
Unmap all subwindows for a window	<code>XUnmapSubwindows(<i>display</i>, <i>w</i>)</code>	Xlib
Unmap popup from within an application	<code>void XtPopdown(<i>popup_shell</i>)</code>	Intrinsics
Unmap widget explicitly	<code>XtUnmapWidget(<i>w</i>)</code>	Intrinsics
Unmap window	<code>XUnmapWindow(<i>display</i>, <i>w</i>)</code>	Xlib
View and edit text widget	<code>XwtexteditWidgetClass</code>	Widgets
Widget manager meta class	<code>XwmanagerWidgetClass</code>	Widgets
Write bitmap to a file	<code>int XWriteBitmapFile(<i>display</i>, <i>filename</i>, <i>bitmap</i>, <i>width</i>, <i>height</i>, <i>x_hot</i>, <i>y_hot</i>)</code>	Xlib

Description	Function	Manual
Write callback procedure	<code>void <i>CallbackProc</i>(w, client_data, call_data)</code>	Intrinsics
Write procedure for an application to register its own action tables with the translation manager	<code>void <i>ActionProc</i>(w, event, params, num_params)</code>	Intrinsics
Write procedure to return resource values from subparts	<code>void <i>get_values_hook</i>(w, args, num_args)</code>	Intrinsics
Write procedure to set almost values for a widget	<code>void <i>AlmostProc</i>(w, new_widget, request, reply)</code>	Intrinsics

## HP X Widgets

The widgets in this table are listed in alphabetical order by widget class name. For more information refer to *Programming With the HP X Widgets*.

Description	Widget Class	Class Name	Subclasses
Description of the interface to constraint resources	Constraint	Constraint	
Core widget meta class	CoreClass	Core	
Arrow drawing widget	XwarrowWidgetClass	Arrow	Core Primitive
Bulletin board manager widget	XwbulletinWidgetClass	BulletinBoard	Core Composite Constraint Manager
Button meta class	XwbuttonWidgetClass	Button	Core Primitive
Popup and pulldown menupane widget	XwcascadeWidgetClass	Cascade	Core Composite Constraint Manager MenuPane
Create a tile suitable for area filling or patterned text	XwCreateTile		
General widget layout manager	XwformWidgetClass	Form	Core Composite Constraint Manager



Description	Widget Class	Class Name	Subclasses
Frame widget	XwframeWidgetClass	Frame	Core Composite Manager
Image editor widget	XwimageEditWidgetClass	ImageEdit	Core Primitive
List manager widget	XwlistWidgetClass	List	Core Composite Constraint Manager ScrolledWindow
Widget manager meta class	XwmanagerWidgetClass	Manager	Core Composite Constraint
Menubutton widget	XwmenubuttonWidgetClass	MenuButton	Core Primitive Button
Menu manager meta widget	XwmenuMgrWidgetClass	MenuMgr	Core Composite Constraint Manager
Menupane meta widget	XwmenupaneWidgetClass	MenuPane	Core Composite Constraint Manager
Menu item separator widget	XwmenuSepWidgetClass	MenuSep	Core Primitive Button MenuBtn
Moves the keyboard focus (and the pointer) to a new toplevel widget	XwMoveFocus	MoveFocus	
Create panels widget	XwPanelWidgetClass	Panel	Core Composite Constraint ManagerClass

Description	Widget Class	Class Name	Subclasses
Popup menu manager widget	XwpopupmgrWidgetClass	PopupMgr	Core Composite Constraint Manager MenuMgr
Primitive widget meta class	XwprimitiveWidgetClass	Primitive	Core
Pulldown menu manager widget	XwpulldownWidgetClass	Pulldown	Core Composite Constraint Manager MenuMgr
Pushbutton widget	XwpushButtonWidgetClass	Pushbutton	Core Primitive Button
Register all of the resource coverters used by the X Widgets	XwRegisterConverters		
Row/column manager widget	XwrowColWidgetClass	RowCol	Core Composite Constraint Manager
Control size of panes widget	XwsashWidgetClass	Sash	Core Primitive
Scrollbar widget	XwscrollbarWidgetClass	ScrollBar	Core Composite Manager
Static image widget	XwstaticcrasterWidgetClass	StaticRaster	Core Primitive
Display static text widget	XwstatictextWidgetClass	StaticText	Core Primitive
Scrolled window widget	XwswindowWidgetClass	ScrolledWindow	Core Composite Manager

Description	Widget Class	Class Name	Subclasses
View and edit text widget	XwtexteditWidgetClass	TextEdit	Core Primitive
Create titlebar	XwttitlebarWidgetClass	TitleBar	Core Composite Constraint Manager
Toggle button widget	XwtoggleWidgetClass	Toggle	Core Primitive Button
Valuator widget	XwvaluatorWidgetClass	Valuator	Core Primitive
Vertical paned manager widget	XwvPanedWidgetClass	VPanedWindow	Core Composite Constraint Manager
Empty window widget	XworkspaceWidgetClass	WorkSpace	Core Primitive



# MANUAL COMMENT CARD

HP Part Number 98594-90070

E1288

Your comments and suggestions help us determine how well we meet your needs.

## X Window System C Quick Reference Guide

	Agree				Disagree
The manual is well organized.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
It is easy to find information in the manual.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The manual explains features well.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The manual contains enough examples.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The examples are appropriate for my needs.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The manual covers enough topics.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Overall, the manual meets my expectations.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### You have used this product:

<input type="checkbox"/> Less than 1 week	<input type="checkbox"/> Less than 1 year	<input type="checkbox"/> More than 2 years
<input type="checkbox"/> Less than 1 month	<input type="checkbox"/> 1 to 2 years	

fold —

Please write additional comments, particularly if you disagree with a statement above. Use additional pages if you wish. The more specific your comments, the more useful they are to us.

Comments: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Please print or type your name and address.

Name: \_\_\_\_\_

Company: \_\_\_\_\_

Address: \_\_\_\_\_

City, State, Zip: \_\_\_\_\_

Telephone: \_\_\_\_\_

Additional Comments: \_\_\_\_\_

X Window System C Quick Reference Guide  
HP Part Number 98594-90070  
E1288



NO POSTAGE  
NECESSARY  
IF MAILED  
IN THE  
UNITED STATES

**BUSINESS REPLY MAIL**

FIRST CLASS

PERMIT NO. 37

LOVELAND, COLORADO

POSTAGE WILL BE PAID BY ADDRESSEE

Hewlett-Packard Company  
CWO PRODUCT MARKETING  
1000 NE CIRCLE BLVD  
CORVALLIS, OR 97330-9988





HEWLETT  
PACKARD

## Errata Sheet

**This Errata Sheet is for HP Part Number: 98594-90070**

**Title:** X Window System C Quick Reference Guide

**Printing Date:** E1288

This errata sheet contains revised information for the *X Window System C Quick Reference Guide* manual to bring it up to date for the current product it documents.

Check the printing history page of your existing manual to verify that it matches the printing date above. If it does not, contact your HP Sales and Service Office for assistance in obtaining a current manual.

Replace the following Intrinsics functions:

Page	Replace This	With This
1-1	XHPAcknowledge ... Intrinsics	XHPAcknowledge ... Xlib
1-43	XtPeekEvent( <i>event_return</i> ) XEvent <i>event_return</i> ;	Boolean XtPeekEvent( <i>event_return</i> ) XEvent * <i>event_return</i> ;
1-55	void XtSetErrorMsgHandler( <i>msg_handler</i> ) XtErrorMsgHandler <i>msg_handler</i> ;	void XtSetErrorHandler( <i>handler</i> ) XtErrorHandler <i>handler</i> ;
1-60	void XtSetWarningMsgHandler( <i>msg_handler</i> ) XtErrorMsgHandler <i>msg_handler</i> ;	void XtSetWarningHandler( <i>handler</i> ) XtErrorHandler <i>handler</i> ;
2-32	void XtAppMainLoop( <i>app_context</i> )	void XtMainLoop()

Hewlett-Packard Company • 1000 NE Circle Boulevard • Corvallis, Oregon 97330-4239

**HP Part Number**  
**98594-90800**

Printed in U.S.A. U0589



98594-90800



Remove the following Intrinsics functions:

Function	Pages	Function	Pages
InsertChild	1-35, 2-1	XtDisownSelection	1-16, 2-35
XCloseDisplay	1-9, 2-15	XtDisplayInitialize	1-17, 2-28
XtAddWorkProc	1-3, 2-34	XtErrorMsg	1-21, 2-3
XtAppAddInput	1-4, 2-34	XtGetResourceList	1-29, 2-24
XtAppAddTimeout	1-4, 2-11	XtGetSelectionValue	1-30, 2-26
XtAppAddWorkProc	1-4, 2-34	XtInstallAccelerators	1-35, 2-28
XtAppCreateShell	1-4, 2-8	XtInstallAllAccelerators	1-35, 2-28
XtAppMainLoop	1-4	XtNewString	1-41, 2-8
XtAppNextEvent	1-4, 2-26	XtOpenDisplay	1-42, 2-31
XtAppPeekEvent	1-4, 2-26	XtOwnSelection	1-42, 2-6
XtAppPending	1-5, 2-6	XtParseAcceleratorTable	1-42, 2-31
XtAppProcessEvent	1-5, 2-38	XtRegisterCaseConverter	1-50, 2-33
XtBuildEventMask	1-5, 2-36	XtSetKeyTranslator	1-57, 2-34
XtCloseDisplay	1-9, 2-6	XtSetSelectionTimeout	1-58, 2-41
XtConvertCase	1-10, 2-13	XtUninstallTranslations	1-65, 2-35
XtCreateApplicationContext	1-11, 2-8	XtUnrealizeWidget	1-66, 2-13
XtDatabase	1-14, 2-24	XtWarningMsg	1-67, 2-3
XtDestroyApplicationContext	1-15, 2-13	XtWidgetToApplicationContext	1-67, 2-17